Università degli Studi di Padova

Department of Information Engineering

Doctor Thesis in

Binary Classifier Inspired by Quantum Detection Theory

Supervisor

Prof. Massimo Melucci

Università di Padova

School Director

Prof. Andrea Neviani

Università di Padova

Doctor Candidate

Prayag Tiwari

First of all, I would like to thank the Almighty God for giving me this opportunity, guiding me during my work, and showering His countless blessings throughout my life. Secondly, I would like to thank my supervisor, Prof. Massimo Melucci, for his continuous support, the advice in the right direction, and trust during all these years of my research work.

I would like to thank especially Dr. Emanuele Di Buccio for the fruitful discussions and his continuous support during all these years.

Many thanks to all my colleagues at the Department of Information Engineering, especially Mr. Qiuchi Li and Mr. Benyou Wang, for the fruitful discussions and enjoyable time spent together at the office.

In the end, I cannot find the words to pay gratitude to all my family members, especially my father and mother, who not only supported me but also became my source of inspiration and motivation.

Prayag Tiwari

Declaration

This thesis's work is based on the research carried out at the Department of Information Engineering, University of Padova, Italy, to fulfill the requirements for the degree of Doctor of Philosophy under the supervision of Prof. Massimo Melucci. No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is completely my own work unless referenced to the contrary in the text.

Prayag Tiwari

Abstract

Decision Theory plays a vital role in the current machine learning field, especially when the optimal decision is required under uncertainty. The application of decision theory is very wide, as it brings statistics, psychology, philosophy, and mathematics together to investigate the decision-making process. On the other hand, quantum theory plays an essential role when the optimal decision is to be made under uncertainty. Quantum Mechanics has shown to be effective as well in several domains where classical theory is not. Such integration of decision theory and quantum mechanics into the field of machine learning can lead to a new research direction.

Machine Learning is an active research area with wide implementations, ranging from speech recognition, strategy optimization, pattern identification and image processing to the investigation of very complex systems. In recent years, the application of Machine Learning (ML) has been used in a wide variety of domains, i.e., medical, transportation, agriculture, etc. Classification is one of the most widely used ML models. There are several existing classification approaches in ML, but classification effectiveness is still an active research area. Ineffectiveness leads to low performance due to the complexity of data. This problem of ineffective performance can be solved using existing traditional approaches or investigations from the other side, which are quantum-inspired approaches. The classification model is proposed from this motivation, which is inspired by the quantum detection model to deal with a diverse range of features, training samples and categories. The proposed classification model is validated on several different datasets and has shown to be effective in many cases compared to the other baselines. The proposed model is also flexible in terms of hyperparameter tuning. Furthermore, macroaverage and microaverage analysis has been done to see the effectiveness of the proposed model. The proposed model has also shown to be efficient in terms of computational cost.

Contents

Abstract v

List of figures x

List of tables xv

Abstract 5

List of Acronyms 5

1 Introduction 11
- 1.1 Background and Motivation . . . 11
- 1.2 Evolution from Classical to Quantum-inspired Models . . . 15
- 1.3 Research Problems . . . 16
- 1.4 Contribution . . . 17
- 1.5 Thesis Organization . . . 18

2 Background 21
- 2.1 Technical Backgrounds . . . 21
  - 2.1.1 Classification . . . 21
    - 2.1.1.1 Support Vector Machine . . . 22
    - 2.1.1.2 Naive Bayes . . . 22
    - 2.1.1.3 K-nearest Neighbours . . . 23
    - 2.1.1.4 Decision Tree . . . 24
  - 2.1.2 Feature Transformation . . . 26
  - 2.1.3 Fundamentals in Quantum Mechanics . . . 27
    - 2.1.3.1 Basic Notations in Quantum Mechanics . . . 27
    - 2.1.3.2 Double Slit Experiment and Quantum Superposition . 29
    - 2.1.3.3 Density Operator . . . 30
  - 2.1.4 Signal Detection Theory . . . 31
- 2.2 Literature Survey . . . 35
  - 2.2.1 Signal Detection Theory for classification . . . 35
    - 2.2.1.1 Classification Images with Signal Detection Theory . . . 35
    - 2.2.1.2 Integration of Signal Detection Theory for Machine Learning classification tasks . . . 37

2.2.2 Effectiveness of Classifier by using Different Machine Learning approaches . . . 37
2.2.2.1 Importance of essential Features in text classification . . 38
2.2.2.2 Focus on Misfit Problem to Improve the Effectiveness . 39
2.2.2.3 Classification Effectiveness followed by a Clustering Approach . . . 40
2.2.2.4 To improve the classification performance using Quantum-Inspired approaches . . . 40
2.2.2.5 Kernel methods Inspired by Quantum Theory for Classification . . . 41
2.2.2.6 Traditional Kernel Methods to Improve Classification Performance . . . 42
2.2.2.7 Neural Networks based models for classification . . . 44

3 Methodology 45
3.1 Introduction . . . 45
3.1.1 Quantum Signal Detection Theory . . . 45
3.2 Quantum Detection Model . . . 48
3.2.1 The Detection Operator . . . 48
3.2.2 Quantum Signal Detection Theory in IR . . . 50
3.2.3 Supervised Learning Framework with the Proposed Quantum Detection Model . . . 51
3.2.3.1 Proposed Quantum Detection Model (QDM) for Classification . . . 52
3.2.3.2 Pseudocode and Data Representation . . . 53

4 Experiments 57
4.1 Introduction . . . 57
4.2 Dataset Description . . . 58
4.2.1 Reuters21578 . . . 58
4.2.2 20Newsgroup . . . 58
4.2.3 MNIST . . . 59
4.2.4 TDT2 . . . 60
4.3 Evaluation Measures . . . 60
4.4 Experimental Setup . . . 61
4.5 Effectiveness Analysis on selected number of Topics . . . 61
4.5.1 Top 9 Topics with at least 100 documents . . . 62
4.5.2 Top Number of selected Topics . . . 63
4.6 Effectiveness Analysis on Range of Features . . . 66
4.6.1 Results on the 20Newsgroup Text Corpora . . . 66
4.6.2 Top Features Visualisation through χ^2 and effect on QDM . . . 69

4.6.3 Results on MNIST handwritten image Dataset 70
4.7 Effectiveness Analysis with the different ranges of Training Samples 80
4.8 Macro-average and Micro-average analysis for effectiveness 88
4.8.1 Description of evaluation parameters 88
4.8.1.1 Macro-average 88
4.8.1.2 Micro-average 89
4.8.2 Analysis on 20Newsgroup Text Corpora 89
4.9 QDM Performance on different Hyperparameter settings 95
4.9.1 For the different value of lambda 95
4.9.2 Effect on QDM performance due to different Detection boundary 95
4.10 Neural Network Autoencoder as Feature Reduction Method to check QDM effectiveness 99
4.11 Comparison of QDM with Neural Network based classification model . . 101
4.12 Efficiency Analysis 103
4.12.1 Computational time on the 20Newsgroup Text Corpora on range of features 103
4.12.2 Computational time on the MNIST Dataset on range of features . 103
4.12.3 Computational time on the 20Newsgroup Text Corpora for each topic 104
4.12.3.1 With Top 100 Features 104
4.12.3.2 With Different training samples 104
4.13 Case study on Failure and Success Analysis of QDM 107
4.14 Computational Complexity or Big $\mathcal{O}$ Analysis 109

5 Discussion **111**

6 Conclusion and Future Works **115**

References **117**

Acknowledgments **129**

Listing of figures

2.1 Support Vector Machine with decision boundary or hyperplane 23
2.2 Naive Bayes visualization 24
2.3 K-nearest Neighbours visualization 25
2.4 Decision Tree visualization 26
2.5 The Double-Slit Experiment: A source S emits light, which can pass through the slit S_1 and S_2, before striking the screen. I_1 and I_2 are the intensities recorded on the screen when S_1 is open, and then when only S_2 is open, respectively. When both slits are open, then the total intensity is no longer equal to the algebraic sum of intensities I_1 and I_2 (source) 30
2.6 Distribution of internal response (internal response indicates an internal impression or state of mind) curve in Signal Detection Theory (SDT) for noise and signal+noise trials. 33

3.1 Classical communication system 47
3.2 Quantum communication system 47
3.3 Proposed Quantum Detection Model (QDM) for Classification 52

4.1 20 Newsgroup Data Visualisation (source) across 20 different newsgroups/topics 59
4.2 Recall Measures of performance for each topic ordered by the measure of the Quantum Detection Model (Quantum Detection Model (QDM)). . . 64
4.3 F-score Measures of performance for each topic ordered by the measure of the Quantum Detection Model (QDM). 65
4.4 Precision Measures of performance for each topic ordered by the measure of the Quantum Detection Model (QDM). 65
4.5 Precision chart based on top selected features from top 5 to top 400 among K-Nearest Neighbors (KNN), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM) and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents precision. 73
4.6 Recall chart based on top selected features from top 5 to top 400 among KNN, DT, NB, SVM and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents Recall. 74
4.7 F-measure chart based on top selected features from top 5 to top 400 among KNN, DT, NB, SVM and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents F-measure. 75

4.8 Top terms Visualisation corresponding to groups, (a) alt.atheism (Topic 1), comp.graphics (Topic 2), comp.os.ms-windows.misc (Topic 4), comp.sys.ibm.pc.hardware (Topic 5), (b) comp.sys.mac.hardware (Topic 6), comp.windows.x (Topic 7), misc.forsale(Topic 8), rec.autos (Topic 9), rec.motorcycles (Topic 10), (c) rec.sport.baseball (Topic 11), rec.sport.hockey (Topic 12), sci.crypt (Topic 13), sci.electronics (Topic 14), sci.med, sci.space (Topic 15), (d) soc.religion.christian(Topic 16), talk.politics.guns (Topic 17), talk.politics.mideast (Topic 18), talk.politics.misc (Topic 19), talk.religion.misc (Topic 20) on the 20 Newsgroups Text Corpora by using χ^2. comp.os.ms-windows.misc(Topic 3) is not visualised because of its high χ^2; furthermore, it was and difficult to visualise features for other topics when the Topic 3 was considered. 76
4.9 Recall chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset. 77
4.10 F-measure chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset. 78
4.11 Precision chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset. 79
4.12 Precision of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction. 82
4.13 Recall of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction. 82
4.14 F-measure of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction. 82
4.15 Precision of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction. 83
4.16 Recall of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction. 83
4.17 F-measure of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction. 83
4.18 Precision of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction. 84
4.19 Recall of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction. 84
4.20 F-measure of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction. 84
4.21 Precision of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction. 85

4.22 Recall of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction. . . . 85
4.23 F-measure of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction. . . . 85
4.24 Precision of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction. . . . 86
4.25 Recall of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction. . . . 86
4.26 F-measure of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction. . . . 86
4.27 Precision of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction. . . . 87
4.28 Recall of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction. . . . 87
4.29 F-measure of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction. . . . 87
4.30 Confusion Matrix of QDM for Topic 1 to Topic 6 on 20Newsgroup Text Corpora . . . 91
4.31 Confusion Matrix of QDM for Topic 7 to Topic 12 on 20Newsgroup Text Corpora . . . 92
4.32 Confusion Matrix of QDM for Topic 13 to Topic 18 on 20Newsgroup Text Corpora . . . 93
4.33 Confusion Matrix of QDM for Topic 19 to Topic 20 on 20Newsgroup Text Corpora . . . 94
4.34 Precision for each topic using QDM (when $\lambda = 0.5$) and SVM. . . . 96
4.35 Recall for each topic using QDM (when $\lambda = 1.5$) and SVM. . . . 96
4.36 F-measure for each topic using QDM (when $\lambda = 1.5$) and SVM. . . . 97
4.37 Threshold chart of QDM by changing the threshold value $\langle x|\Delta|x\rangle > 0.3$, 0.5, 0.6, and 0.7 on 20 Newsgroup Dataset. X-axis represents Topics and Y-axis represents Precision, Recall, F-measure in these 3 charts. . . . 98
4.38 QDM Performance on different parameters. `QDM_100_1K` means parameters consist of a hidden-size of 100 with 1000 maximum number of training epochs or iterations. `QDM_50_1K` means parameters consist of a hidden-size of 50 and 1000 epochs. `QDM_25_1K` means parameters consist of a hidden-size of 25 and 1000 epochs. `QDM_10_1K` means parameters consist of a hidden-size of 10 and 1000 epochs. `QDM_10_2K` means parameters consist of a hidden-size of 10 and 2000 epochs. `QDM_25_2K` means parameters consist of a hidden-size of 25 and 2000 epochs. . . . 100
4.39 Computational time of KNN, DT, NB, SVM, and QDM on 20 Newsgroup Text Corpora for each topic with 100 features . . . 105

4.40 Computation time of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction 106
4.41 Computation time of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction 106
4.42 Computation time of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction 106

Listing of tables

2.1 Summary of decision about the presence or absence of signal. For each actual state and decision, the corresponding outcome has costs. 32

4.1 Hardware settings for QDM. 61
4.2 Hardware settings to train Autoencoder (when QDM is trained with NN autoencoder) 61
4.3 Classification performance by Naive Bayes 62
4.4 Classification performance by Support Vector Machine 63
4.5 Classification performance by Quantum Detection Model (QDM) 63
4.6 Precision Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 68
4.7 Recall Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 68
4.8 F-measure Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 69
4.9 Recall Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400. 71
4.10 F-measure Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400. 72
4.11 Precision Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400. 72

4.12 Macro-average and Micro-average estimates where Pr^m denotes macro-average precision, R^m denotes macro-average recall, Pr^μ denotes micro-average precision, and R^μ denotes micro-average recall. 90
4.13 Performance Comparison of QDM with neural based models for topic categorization on 20Newsgroup Text Corpora. 102
4.14 Computation Time in seconds for KNN, DT, NB, SVM, and QDM on the 20 Newsgroup Text Corpora 103
4.15 Computation Time in seconds for KNN, DT, NB, SVM, and QDM on the MNIST Handwritten Image Dataset 104

Abstract

Decision Theory plays a vital role in the current machine learning field, especially when the optimal decision is required under uncertainty. The application of decision theory is very wide, as it brings statistics, psychology, philosophy, and mathematics together to investigate the decision-making process. On the other hand, quantum theory plays an essential role when the optimal decision is to be made under uncertainty. Quantum Mechanics has shown to be effective as well in several domains where classical theory is not. Such integration of decision theory and quantum mechanics into the field of machine learning can lead to a new research direction.

Machine Learning is an active research area with wide implementations, ranging from speech recognition, strategy optimization, pattern identification and image processing to the investigation of very complex systems. In recent years, the application of ML has been used in a wide variety of domains, i.e., medical, transportation, agriculture, etc. Classification is one of the most widely used ML models. There are several existing classification approaches in ML, but classification effectiveness is still an active research area. Ineffectiveness leads to low performance due to the complexity of data. This problem of ineffective performance can be solved using existing traditional approaches or investigations from the other side, which are quantum-inspired approaches. The classification model is proposed from this motivation, which is inspired by the quantum detection model to deal with a diverse range of features, training samples and categories. The proposed classification model is validated on several different datasets and has shown to be effective in many cases compared to the other baselines. The proposed model is also flexible in terms of hyperparameter tuning. Furthermore, macroaverage and microaverage analysis has been done to see the effectiveness of the proposed model. The proposed model has also shown to be efficient in terms of computational cost.

Listing of acronyms

ML Machine Learning

DT Decision Theory

QM Quantum Mechanics

SDT Signal Detection Theory

2AFC Two-Alternative Forced-Choice

PDF probability density function

QDM Quantum Detection Model

SVM Support Vector Machine

IR Information Retrieval

DF Document Frequency

KNN K-Nearest Neighbors

NB Naive Bayes

DT Decision Tree

NN Neural Networks

ASM Attribute Selection Measures

LSA Latent Semantic Analysis

BFGS Broyden–Fletcher–Goldfarb–Shannon

MBPNN Modified Backpropagation Neural Network

RELM Regularization Extreme Learning Machine

LLSF Linear Least-Square Fit

CBC Clustering-based Classification

BGSA Binary Gravitational Search Algorithm

LOCV Leave-One-Out Cross-Validation

RELM Regularization Extreme Learning Machine

RBF Radial Bias Function

RKHS Reproducing Kernel Hilbert Space

POVM Positive Operator-Valued Measure

PLS Partial Least Squares

SVC Support Vector Classifier

PCA Principal Component Analysis

HSIC Hilbert-Schmidt Independence Criterion

SOTA state-of-the-art

SNR signal-to-noise ratio

NLP Natural Language Processing

PDF Probability Density Function

ANN Artificial Neural Network

CFN complex-valued fuzzy network

DBN Deep Belief Network

CNN Convolutional Neural Network

LSTM Long Short-Term Memory

CHAPTER 1

Introduction

1.1 Background and Motivation

Decision Theory [22] provides a general architecture to study and explain the decisions that are made in ambiguous or uncertain situations. For decades, it has been applied in psychophysics; the study of the connection between physical stimulus and its psychological effects. The application of decision theory includes almost all inductive sciences aw well as many normal circumstances that people face in their everyday lives when there is uncertainty to make a possible decision. Let's consider a very common situation in which a person needs to decide whether some condition is present. Such types of decisions are easier to make when the evidence and the alternatives are clear. It is often the case that the alternatives are different, but there is ambiguity in the evidence which is used for decision making. Here are two examples:

- A doctor is trying to examine a patient and make a diagnosis. The patient shows a number of symptoms, and the doctor needs to decide whether a specific disorder is present or not. Due to ambiguity, some symptoms may point away from the disorder, while others point towards the disorder. Furthermore, the patient cannot give a clear description of the symptoms since the patient is very confused. In this case, diagnosing correctly is not obvious.

- A witness from a crime scene is presented with the suspects and asked to identify them. The witness tries to recall the incident, but his memory is not so clear. For the witnesses, the crime scene was dark, stressful and difficult, so he confuses the suspects. Furthermore, the witness has already been interviewed by others regarding the inci-

dent. Two alternatives are possible: either the suspect was at the crime scene or not, but the decision is ambiguous based on the information that the witness brings.

In the above examples have something in common. In the decision making process in both cases, the responses are yes/no answers, but the information used in the decision making is ambiguous and incomplete. Such limitations bring difficulty to decision making. So, the error can occur, no matter how careful decision-makers are. It is quite difficult to have a full understanding of the decision making process under some circumstances, for example, in the two cases above. There are many particulars that decisions depend upon, such as decision-maker knowledge, her/his beliefs and expectations, subsequent explanations of the original observations, and the like. Domain-specific knowledge is required for a better understanding, for example, in the case of seismology and medical diagnosis. Some features of yes/no decisions surpass the specific circumstances in which a decision is made. These general elements are treated by a theory called *signal detection theory*, which is often used in the situation of decision making under uncertainty to reduce the uncertainty level. Further details about this theory can be found in the next chapters as well as in these works [10, 37, 94]. The above cases make the decision hard to make and the decision is often intrinsically ambiguous. SDT provides an effective solution for these cases.

The application of SDT is very wide and has been used widely in many domains. For example, the classification of images has been an essential component for exploration in visual psychophysics, and most of the theory to understand the classification of images is based on traditional models from SDT. Some related works [75, 6, 77, 76] about the classification of images with SDT can be found in Section 2.2.

The main focus of this thesis is a classification task since it is a typical decision-making scenario. **The connection between detection and classification** can be understood from the signal processing domain where the issue appears whenever a decision is to be made among distinct hypotheses regarding a received waveform. The SDT framework allows us to decide whether the obtained waveform comprises of *signal with noise* or *only noise*. The signal classification framework enables us to determine whether the detected signal corresponds to the predefined classes of signals. The main aim of SDT with classification is to provide structured strategies for constructing such frameworks that allow to reduce the average number of decision errors. The fundamental ground of SDT with classification can be found in the statistical decision theory which is known as *hypothesis testing*.

Quantum Mechanics (QM) plays an essential role in the notion of *uncertainty*. QM has also been applied outside of physics to studies like, quantum chemistry, quantum cryptog-

raphy, quantum biology , quantum cognition, and quantum social science. QM deals with the microscopic world, which is far smaller than anything that directly effects our sensory perceptions. It is essential to convey information about the quantum system to the gross macroscopic world in which human senses operate. QM has also been widely used to deal with uncertainty in many other domains. For example, many studies in cognitive science show that probabilistic outcomes of human decision-making under uncertainty do not always follow the law of classical probability theory. So, the following question arises: what type of probability theory would be the better alternative to explain those decisions made under uncertainty? The probabilistic model drawn from QM has shown to be a better alternative and is increasingly gaining attention in the cognitive science domain. The logic underlying Kolmogorov's probability theory is based on Boolean algebra due to its set-theoretic structure. The events can always be combined through logical conjunction according to the associated Boolean logic. Since logical conjunction is commutative, stating the combined event 'A and B' is the same as expressing 'B and A'. More formally, event order does not matter according to the commutative property. In contrast, it is not necessarily true that the sequence of event 'A and B' is the same as the sequence of 'B and A', which follows the property of non-commutativity by von Neumann's machinery of projective geometry [108]. Such violations have also been exploited in the case of multidimensional relevance judgment in Information Retrieval (IR) and modeled using QM in the work [106]. In the proposed work, topicality, understandability, and reliability are considered as three dimensions of relevance. The users were posed with three questions based on topicality, understandability, and reliability in different orders. They were asked to respond in "Yes" or "No". It was found that asking the user questions in different orders led to different results (known as order effect). A quantum-like structure for relevance judgment was proposed in the work. It was found that quantum probability theory is a better way to model multidimensional relevance judgment than classical models from the experimental data [106].

Another work [72] in the quantum cognition domain proposes a decision framework using quantum probability theory to explain the probabilistic nature of decision-making processes to model the human paradox and irrational decisions. The proposed work uses a quantum-like Bayesian network formalism to comprehend the idea of maximum expected utility along with the concept of quantum-like influence diagrams. The proposed work also uses expected utility theory as it was used in the previous works, but the quantum interference terms are obtained in a quantum-like Bayesian, which is considered advantageous because it influence the probabilities utilized to estimate the expected utility of decision making

[72].

It can be seen that QM can be useful in such scenarios. The integration of QM with SDT formalizes a new theory and can be very helpful in dealing with uncertainty in decision-making. This reformulation of QM and SDT is known as *quantum* SDT. Quantum SDT is the reformulation of statistical decision theory to detect signals in random noise. More details about quantum SDT [48, 45, 47, 46] can be found in the next sections.

The theoretical framework of decision theory and quantum SDT can be very effective in the ML domain as well. Decision Theory [22] has been used in ML as a sound foundation for making decisions under uncertainties. ML models are usually developed with the focus of learning and predictions. Machine Learning theory is an essential part of both statistics and artificial intelligence, and its source can be found at the start of artificial intelligence in the 1950s. Arthur Samuel defined ML as the "field of study that gives computers the ability to learn without being explicitly programmed" [86, 88].

Machine Learning is a well-known research domain with broad implementations, including speech recognition [54], strategy optimization, pattern identification, Natural Language Processing (NLP) [110, 61, 103, 127, 128, 116], image processing [122, 115, 43], and the investigation of very complex systems. ML also has a vast range of applications, for example, iris recognition for security systems or person re-identification [122, 115], assessment risk in the financial market, consumer behavior analysis, and so on. In general, machine learning is applied whenever we require computers to analyze data based on the provided experience. This typically includes previously collected data and efficient machine learning models that can process and analyze a large amount of data. ML has been implemented successfully in many fields and will be even more important in the near future. Machine learning algorithms fall into three different categories: supervised, unsupervised, and reinforcement learning. This work is about classification framework [102, 101, 100, 28, 99], which is the group of supervised machine learning approaches.

Classification, an approach in ML and IR, has been an active research area for years. The application of classification is extensive in several domains. Classification is the key task of ML in several artificial intelligence fields, i.e., IR [66, 68, 65, 70, 69], data mining, recommender systems, NLP, computer vision, and so on. The main objective of the classification approach is to predict the target for some given data points. Classification can be binary (either '0' or '1'), multi-class (more than 2 categories), or multi-label classification (one sample may belong to more than one label). It is also important to mention that multi-class and multi-label classification problem is generally solved using binary classifier by decomposing

it into binary classification problem. There are several existing classification approaches, but their effectiveness has been an active research question. The effectiveness (performance related to the accuracy, precision, recall, and f-score) and efficiency (computation time and complexity) of classifiers depend on the complexity of the data (diverse range of features and training samples for the categories), flexibility in hyperparameter tuning, and complexity of the algorithm.

1.2 Evolution from Classical to Quantum-inspired Models

There are several existing works that discuss classification effectiveness. The effectiveness of classifiers is generally improved by using some traditional ML approaches and further related works can be found in Section 2.2. Decision theory plays an essential role in the classification, which is the backbone of this work, so it is essential to discuss this theory's application in different domains. Section 2.2.1 shows how detection theory is helpful for the classification images, and how the integration of SDT with ML classification can help to solve different tasks. Furthermore, the classifier's effectiveness is improved by using different ML approaches (i.e., feature selection, focusing misfit problem, clustering approach, neural networks, kernel method for classification, etc.), which is discussed in Section 2.2.2.

However, it is necessary to mention what makes the framework proposed in this thesis different from other models. Existing ML models are based on the classical probability theory [13, 73]. With the increasing complexity of data, existing ML models are less effective, and the need for more advanced algorithms becomes essential. Firstly, limited works explore the use of quantum-inspired frameworks because most ML models are based on classical probability theory. QM may give rise to the development of a more advanced algorithm because it has a higher degree of freedom than classical theory, and QM is rich in mathematical formalism. The strength of QM has already been demonstrated for information processing by physicists [63, 29]. This quantum mechanical structure can also be inspiring and motivating to develop ML models. So, it is feasible to use quantum probability theory by replacing the classical one with advanced quantum-inspired classification models. The inspirations from QM and SDT with classification allow us to investigate the new direction of classification.

1.3 Research Problems

The whole idea of building classification models is to predict human decisions in real scenarios. The effectiveness of classifiers is still an active research problem in many domains. Classifiers' ineffective performance is generally due to the diverse range of features, training samples, and categories. The ineffective performance of the classifier also depends upon the non-flexibility of hyperparameter tuning. Therefore, the key challenge in classification is how to obtain optimal effectiveness and efficiency despite the complexity of data [21, 58, 5, 82, 93, 53, 64, 4]. Therefore, this dissertation aims at answering the following research questions:

RQ1: How can classifiers obtain effectiveness despite the issue of the *diverse range of features and training samples*? More specifically, a classifier maybe effective for one category but ineffective for others depending upon the feature extraction quality from the data and the required training samples.

Example supporting RQ1: For instance, ImageNet (a very large image dataset consisting of 1000 categories, $1.2M$ training sets, and $100K$ test images) is one example where existing classifier performance is not so effective, while Reuters-21578 (Reuters Text Categorization Dataset consisting of 135 categories, 13625 training sets, 6188 test sets) is an another example where classifier performance is effective for one category but ineffective for others due to diverse categories and high dimensionality of the data. Theoretically, high dimensional space allows more information to be stored, but in practice real world data consist of highly irrelevant features which lead to a sparsity issue. The effectiveness of a classifier improves with the increasing number of features up to some top optimal feature size, but when the number of features keeps increasing without increasing the size of the training samples, it leads to classification ineffectiveness with low performance.

RQ2: Obtaining high macro-average metric is still a challenging task with existing classification models especially when the one-vs-all strategy is used, so how can effectiveness be achieved for classifiers?

RQ3: Hyperparameter flexibility is still an open issue to some degree with the existing classification models, so what can be done to make the classification model more effective in

terms of hyperparameter tuning?

RQ4: How can optimal efficiency be obtained in classification despite the complexity of data?

1.4 Contribution

The main contribution of this thesis is to tackle the challenges of classification ineffectiveness due to the diverse range of features, training samples and categories, which is a very generic problem (as mentioned in RQ1 and RQ2) in ML. Hyperparamater flexibility is another issue (as mentioned in RQ3) with the existing classification models to some degree because hyperparamater tuning allows to improve the model performance. There are two possible approaches to solve these issues. The first approach is to use several existing classification algorithms to determine which algorithm works best, which is a more general way. The second approach is to investigate an algorithm that is fundamentally different from existing ones, which can in principle better handle the diverse range of features, training samples, categories, and allow for more flexibility in terms of hyperparamater tuning. The second approach is the main focus of this work, where quantum SDT is used for the proposed classification model instead of classical theory. To tackle the challenges of the classifier (as mentioned in RQ1, RQ2, RQ3, and RQ4) due to the complexity of data (a diverse range of features, training samples, and categories) and flexibility in hyperparameter tuning, a quantum-inspired classification model is proposed and experimented [102, 101, 100, 28], which is the extended work of the quantum-inspired framework for relevance feedback [67]. The proposed model outperforms the existing models for many categories where others can not perform well due to a diverse range of features and training samples, as mentioned in RQ1. It outperforms baselines in terms of macro-average precision (as mentioned in RQ2), and it has more flexibility in terms of hyperparameters tuning (as mentioned in RQ3). The proposed model also improves computational cost compared to a few other models (as mentioned in RQ4).

It is worthwhile to mention that SDT can provide structured strategies for constructing such frameworks that empirically reduce the average number of decision errors [37, 10, 75]. The theoretical motivation of quantum SDT is that it can provide better results than the classical one, which is also reported in some works about communication systems [45, 47, 46]. The main advantage of QDM is that it allows to leverage more effective signals as these signals are less susceptible to classification errors. This optimal probability of detection or reducing

the probability of error allows to classify more accurately.

During the training process, top features are selected using feature selection corresponding to some given categories. These top features allow the model to be trained with reduced errors and better performance during the prediction phase. Several datasets are used, which consist of a range of features, training samples, and categories. Comprehensive experiments have been performed for the proposed quantum-inspired classification model on a range of features (top selected features using the feature selection approach) and diverse training samples (the model is trained with different training sets) to provide better insight for the number of categories. A quantum-inspired classification model achieved effectiveness to some extent while other models could not for several categories. Furthermore, overall classifier performance is estimated using macro-average and micro-average analysis (as mentioned in RQ2). The proposed model also outperformed baselines in terms of macro-average precision. The proposed model has also have shown to be effective in terms of hyperparamater tuning. This work could inspire further investigation into classification from the quantum side.

1.5 Thesis Organization

The organization of this thesis is as follows:

- **Chapter 2** provides a brief summary of related works. The proposed model uses decision theory, SDT, QM, quantum SDT, and ML. There are several works which also use SDT for classification so some surveyed works are also presented in the section. Furthermore, I will outline how different approaches have improved the classification performance, for example how essential features improve the classification effectiveness, how focusing on the misfit problem leads to better classification performance, and how a clustering approach can improve performance. I will also point out how some quantum inspired approaches are used to make classification approaches effective, how neural network models play an essential role in improving classification, and how the kernel method proves to be effective for the SVM classifier. Section 2.1 discusses fundamental details about the different notions used in this thesis.

- **Chapter 3** discusses the proposed model inspired by quantum signal detection theory. Section 3.1.1 provides some differences between classical and quantum communication frameworks regarding how a coder plays an essential role. Section 3.2.3 provides the general framework of a supervised learning framework. The details regarding QDM for classification can be found in these Sections. The pseudocode of QDM can be found in Section 3.2.3.2.

- **Chapter 4** provides the experimental details. The dataset description in detail can be found in Section 4.2. The details about evaluation measures can be found in Section 4.3. Several different datasets were used to evaluate the performance of the proposed model along with the baselines. The details about the experimental setup can be found in Section 4.4. Furthermore, results on different datasets based on the range of features and different training samples used to observe the effectiveness of the proposed model can be found in Sections 4.5, 4.6, and 4.7. Macro-average and micro-average analysis has also been done, which can be found in Section 4.8. Section 4.9 discusses the hyperparamater tuning. The proposed QDM for classification was also trained with a neural network autoencoder and the obtained results are reported in Section 4.10. QDM is also compared with Neural Networks (NN) based classifier in Section 4.11. The efficiency analysis can be found in Section 4.12. Lastly, a case study has been done in order to give insight into some specific topics and provide details about the failure and success of QDM in Section 4.13. Big $\mathcal{O}$ Analysis is also done in Section 4.14 in order to see the complexity of the proposed model as well as the baselines.

- **Chapter 5** provides the discussion about the proposed work. It also discusses the possibility to use the proposed model in other domains for different tasks.

- **Chapter 6** provides the conclusion of the proposed model followed by future works.

CHAPTER 2

Background

2.1 Technical Backgrounds

The main aim of this Section is to provide the technical notions in order to understand the notion of QDM.

2.1.1 Classification

Classification is the key task in machine learning, i.e., information retrieval, data mining, recommender systems, computer vision, and so on. It is the way of predicting the target of some given data points. Targets are sometimes given different names, i.e., classes, categories, or labels. For example, in spam detection where email service providers can be identified as spam or not spam. This is the binary classification task where spam and not spam are two classes. The classifier requires some training data to understand how a given input feature relates to the class. In the spam detection example, spam and not spam emails can be used as training data. If the classifier learns precisely from the input data, then it can classify unknown emails.

The classification is supervised learning tasks where the target variable is given. Generally, the classification task can be divided into three types.

- *Binary classification* task, where the given sample belongs to one of the two classes (0/1, Yes/No, +/-).
- *Multi-class classification* task, where each sample belongs to only one of three or more than three classes, for example, classifying a set of fruit images which may be apples, orange, or pears.

- *Multi-label classification* task, where each sample belongs to multiple labels. For example, a text sample may belong to multiple labels (finance, religion, education, or politics) at the same time.

Generally, the multi-class [99] and multi-label classification problem is solved using a binary classifier, where the multi-class and multi-label problems are decomposed into binary.

2.1.1.1 Support Vector Machine

SVM is a supervised ML approach used for classification and regression tasks, although SVM is widely used for the classification task. SVM's primary function is to find the best hyperplane or decision boundary in the feature space that distinguishes the data points. There could be many hyperplanes or decision boundaries that separate the two classes of data points, but it is necessary to find the place with maximum margin. The marginal distance must be maximized to ensure that unseen data points are classified with high confidence. SVM can be understood visually from Figure 2.1. In terms of adaptability, the set of feature vectors is used in SVM, instead of event space as in NB. A linear classifier with maximum-margin hyperplane was proposed by Vapnik in 1963 [107]. Later, Vapnik and his colleagues proposed to construct a non-linear classifier by using a kernel function.

2.1.1.2 Naive Bayes

Bayes' theorem proposed a formalism to estimate the posterior probability, $P(y|x)$ as follows:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

where, x, y are events of some probability space, $P(y)$ denotes the prior probability of the target class, $P(y|x)$ denotes the posterior probability of the class for the feature vector, $P(x|y)$ denotes the likelihood of the feature vector observed in some given class, and $P(x)$ denotes the prior probability of feature vector [74].

Naive Bayes classifier assumes that independency exists among attributes for every class to minimize the computational burden. Such an assumption is known as conditional independence of class and it can be expressed in the following equation:

$$P(x|y) = \Pi_{i=1}^{d} P(x_i|y)$$

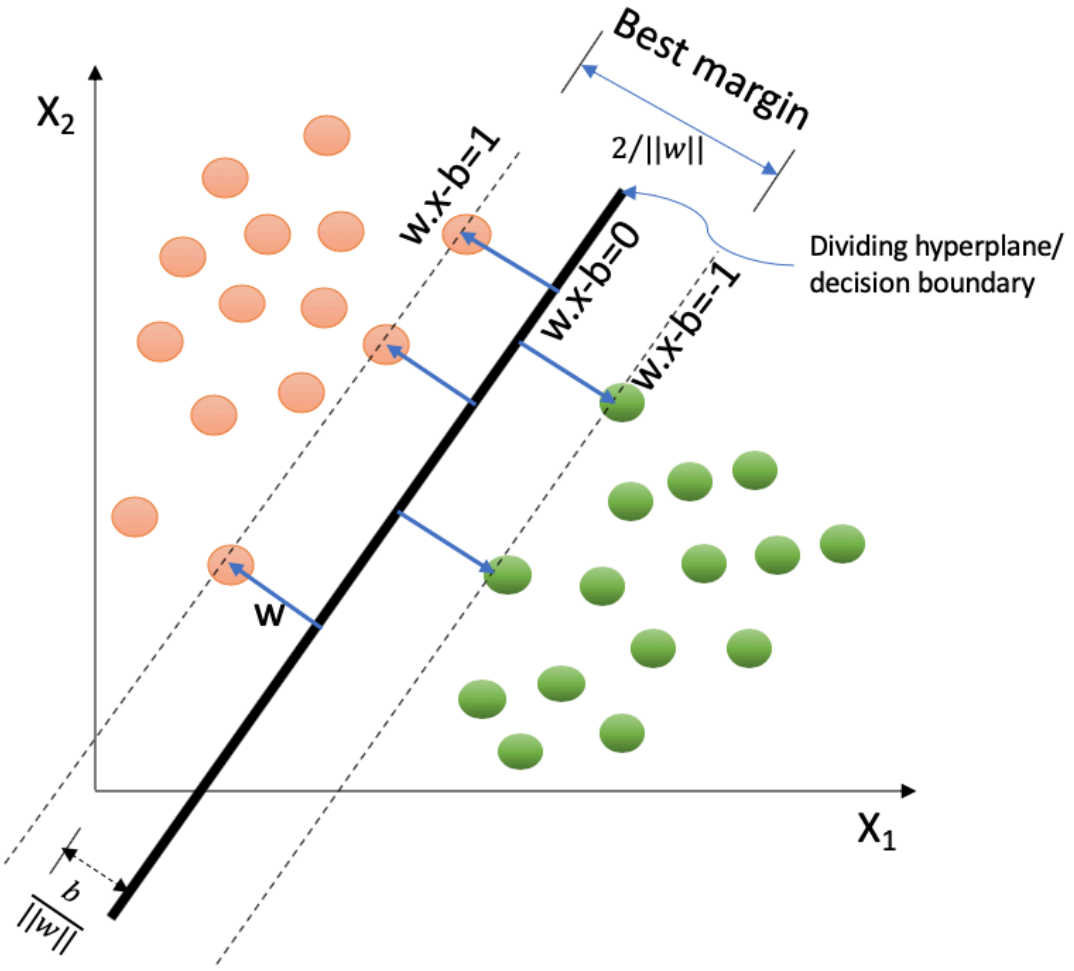


Figure 2.1: Support Vector Machine with decision boundary or hyperplane

where d represents some dimension of feature space, and x_i is the i^{th} feature of vector $x \in \mathbb{R}^d$. NB can be understood from Figure 2.2 as well.

2.1.1.3 K-nearest Neighbours

KNN is a supervised ML approach that is used for classification and regression tasks. KNN supposes that similar things exist near each other. KNN utilizes a feature similarity (distance function or similarity function) to predict the new data point values. This algorithm has been in use since the 1970s for several pattern recognition tasks. A data point is classified by the most votes of its nearest data points, with the data point being assigned to the class which is most common among its K nearest neighbors. There are several distance function

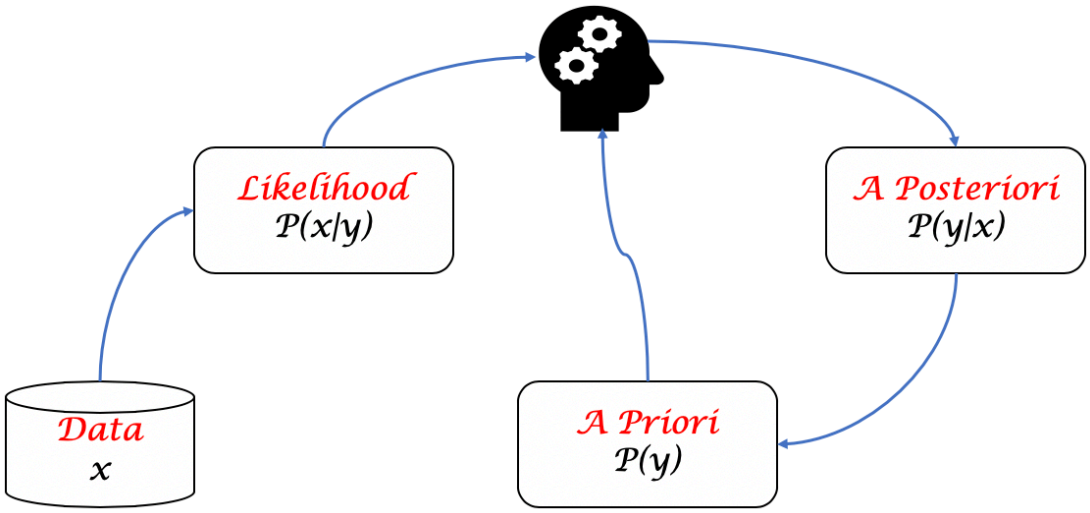


Figure 2.2: Naive Bayes visualization

used in KNN , and few of them can be found as follows:

- Euclidean distance function, $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
- Manhattan distance function, $\sum_{i=1}^{k}|x_i - y_i|$
- Minkowski distance function, $(\sum_{i=1}^{k}(|x_i - y_i|)^q)^{\frac{1}{q}}$

KNN can be seen in Figure 2.3 where a new example is finding the neighbors and voting for labels.

2.1.1.4 Decision Tree

The decision tree is used for classification as well as regression tasks. DT is like a flowchart tree architecture, where nodes denote a feature, edges represent a decision rule, and each leaf node represents the outcome. The node at the top is known as the root node, where it learns to split the data based on the feature values. Recursive partitioning is used to partition the tree. DT can be explained in the following steps:

- Select the top essential feature using Attribute Selection Measures (ASM) (Information Gain or Gain Ratio or Gini Index) to partition the records.

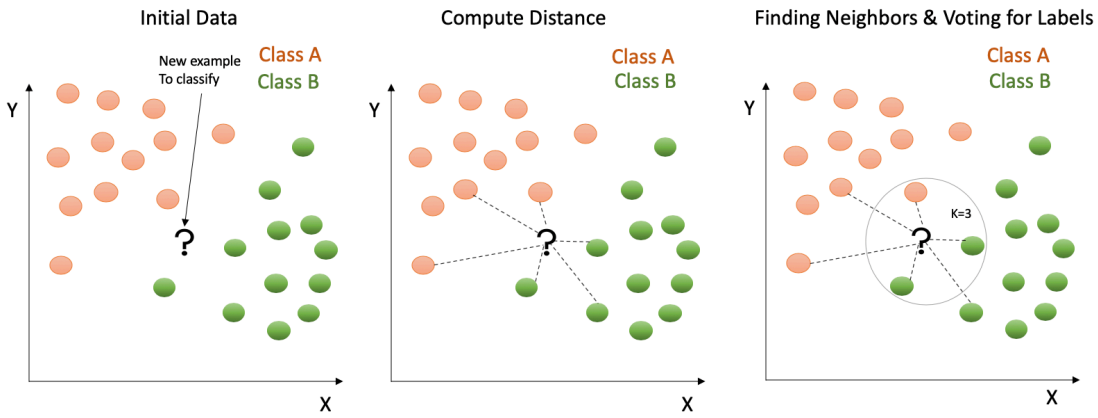


Figure 2.3: K-nearest Neighbours visualization

- This feature is considered a decision node. Then, the dataset is partitioned into smaller subsets.

- Finally, the tree is constructed by repeating these steps in a recursive manner for every child until one of the given conditions is matched as follows: (1) all the tuples correspond to the same feature value, (2) there are no samples left, (3) there are no features left.

Information Gain is explained as the decrements in entropy $E(D)$, and it refers to the rate of the amount of uncertainty in the given set D (data):

$$E(D) = \sum_{c \in C} -P(c) \log_2 P(c) \tag{2.1}$$

where $P(c)$ is some probability that an arbitrary tuple in dataset D corresponds to class $C \in \{0, 1\}$.

Information gain $info(D, F)$ describes what label of uncertainty is reduced in D after partitioning D for feature F:

$$info(D, F) = E(D) - E(D, F) \tag{2.2}$$

This Eq. 2.2 can be extended and written as follows:

$$info(D, F) = E(D) - \sum_{t \in T} P(t)E(t), \tag{2.3}$$

where T is the subset from the partition, $P(t)$ is the the number of features ratio in t to the number of features ratio in D set, and $E(t)$ is the entropy of t subset. DT can be understood from the example as s person being fit or unfit in Figure 2.4.

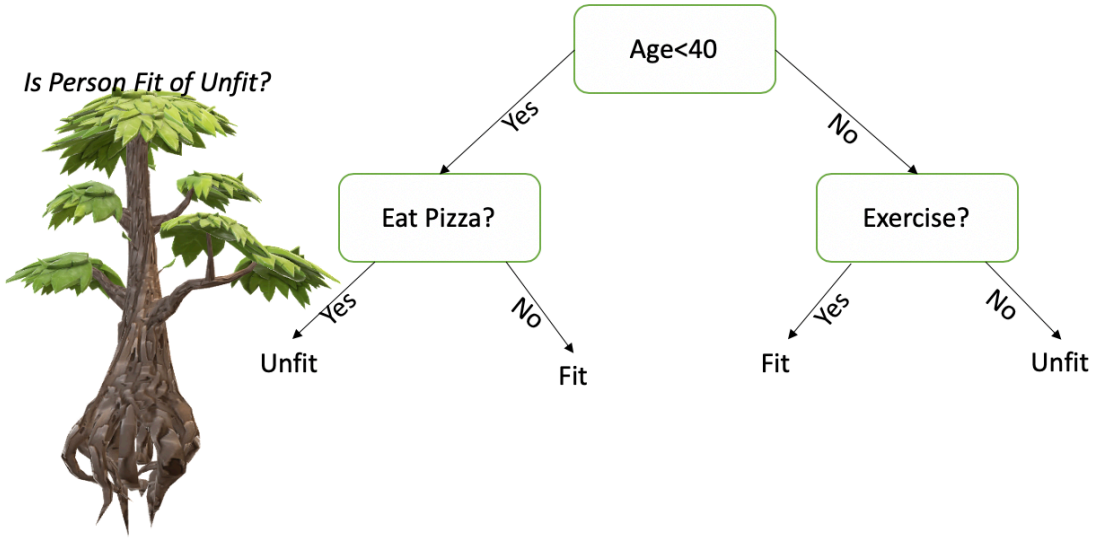


Figure 2.4: Decision Tree visualization

2.1.2 Feature Transformation

Feature transformation is an essential step during data processing. Generally, the dimensions of data are high, and there are many irrelevant features. Those irrelevant features make ML models weak in terms of performance. Feature selection is often used to transform high dimensional data into low dimensional data by providing more relevant features. This step automatically improves performance and efficiency as well. There are several feature selection models, but χ^2 is widely used.

χ^2 is used widely in statistics to test the independence of two distinct events. This approach is mainly used for the categorical attributes in a dataset. χ^2 is computed between each attribute and target. It selects the number of desired attributes with the best χ^2 scores. The expected count of E and observed count O can be computed from two variables' given data. The main aim of χ^2 is to calculate the deviation between expected count E and observed count O.

$$\chi^2 = \sum_{i=1}^{n} (O_C - E_C)^2 \,/\, E_C, \tag{2.4}$$

Where, E_C denotes the number of expected observation in class C, O_C denotes the number of observation in class C.

2.1.3 Fundamentals in Quantum Mechanics

At the start of the 20^{th} century, Maxwell and Newton's existing laws failed to explain somehow when some challenges came up to understand the observed results of some specific experiment. In the conclusion, a novel mathematical framework known as quantum mechanics was born, and novel theories of physics known as quantum physics originated in this framework.

QM developed over many decades, starting with set of controversial mathematical statements of some experiments where classical theory failed to explain those controversies. Several scientists contributed to the foundation of three revolutionary principles (Quantized properties, Particles of light, and Waves of matter) that slowly gained the attention and experimental verification from 1900 to the 1930s. The foundation was initiated by Werner Heisenberg and Erwin Schrodinger and later reached maturity in the 1920s and 1930s. In 1926, Schrodinger proposed the fundamental equation known as Schrodinger's equation for wave mechanics, as shown in the following equation:

$$H\psi = E\psi \tag{2.5}$$

The eigenfunction ψ describes the system's state, E denotes the eigenvalue for the system energy, and H represents the Hamiltonian operator.

Heisenberg introduced the *uncertainty principle*, stating that the position and momentum of any particle (i.e., electron) cannot be measured simultaneously at arbitrary precision [44], thus establishing the principle that nature is uncertain. The measurement of macroscopic world entities are deterministic according to the classical theory because a system state can be absolutely measured several times; randomness is an intrinsic feature of the invisible world according to QM. This uncertainty does not occur because of the imprecision in measurement, it is rather due to the intrinsic randomness of the state of the system [16].

2.1.3.1 Basic Notations in Quantum Mechanics

The Dirac, or "bra-ket" notation is adopted in QM. In classical theory, column vectors represent the states. In QM, column vectors are written compactly and linearly. For example, a k-dimensional column vector A

$$A = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{k-1} \end{pmatrix}$$

can be represented in terms of ket notation as $|A\rangle$ in the following way

$$|A\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{k-1}|k-1\rangle,$$

where $|0\rangle, |1\rangle \ldots$ are basis vectors with the following representation

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, , |1\rangle = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, |k-1\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

The bra notation of the same vector, $\langle A|$, corresponds to the conjugate transpose of the ket representation:

$$|A\rangle = \langle A|^\dagger$$

This means that the inner product of vector A can be expressed as

$$\langle A|A\rangle = \begin{pmatrix} \alpha_0^* & \alpha_1^* & \ldots & \alpha_{k-1}^* \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} = |\alpha_0|^2 + |\alpha_1|^2 + \cdots + |\alpha_{k-1}|^2 .$$

In the same manner, the outer product of vector A, which is also known as the *quantum Projection operator*, can be represented as

$$|A\rangle\langle A| = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} \begin{pmatrix} \alpha_0^* & \alpha_1^* & \ldots & \alpha_{k-1}^* \end{pmatrix} =$$

$$\begin{pmatrix} |\alpha_0|^2 & \alpha_0\alpha_1^* & \dots & \alpha_0\alpha_{k-1}^* \\ \alpha_1\alpha_0^* & |\alpha_1|^2 & \dots & \alpha_1\alpha_{k-1}^* \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{k-1}\alpha_0^* & \alpha_{k-1}\alpha_1^* & \dots & |\alpha_{k-1}|^2 \end{pmatrix}, \text{ where } |\alpha_0|^2 + |\alpha_1|^2 + \cdots + |\alpha_{k-1}|^2 = 1$$

The Dirac notation allows the explicit labeling of the basis vectors, which play an essential role in QM and the quantum-like model used in this thesis.

2.1.3.2 Double Slit Experiment and Quantum Superposition

Young's double-slit experiment demonstrates the distinctive feature of quantum mechanics. The nature of light has been a crucial discussion over the years. Newton believed that an object's sharp shadow could not be described if light were the wave. So, he developed the idea that light is not a wave like sound, which can be heard behind objects. Nevertheless, in the nineteenth century, the interference experiment was demonstrated as a wave-like nature of light. The best way to introduce interference is to refer to the Feynman lecture on physics [31].

Young developed the superposition principle: if two waves, ejected by a single source, collapse on the screen, their amplitude adds up algebraically. This property, known as interference fringes, was first observed in the double-slit experiment shown in Figure 2.5. Light is emitted by source S, travel through the two slits, S_1 and S_2, and illuminates a screen. The interference pattern can be seen in Figure 2.5. The main point from the given experiment is that the intensity I on the screen is distinct from the algebraic sum of intensities I_1 and I_2 generated by both slits individually; that is, we close slit S_1 or S_2, respectively [11]. Therefore, we have as follows:

$$I \neq I_1 + I_2 \tag{2.6}$$

The superposition in QM is one of the core ideas that makes QM strong [31]. This superposition theorem can be expressed by the following mathematical equation:

$$|\psi\rangle = \alpha_1 |\phi_1\rangle + \alpha_2 |\phi_2\rangle \qquad \text{where, } |\alpha_1|^2 + |\alpha_2|^2 = 1. \tag{2.7}$$

In other words, any state vector with some specific state ψ exists at both state ϕ_1 and ϕ_2 simultaneously. The particle was in a superposition state (existing at both state ϕ_1 and ϕ_2) before measurement, but when the measurement is taken, the particle state vector collapses at one of the two-state ϕ_1 and ϕ_2. This collapse of the state vector to any basis cannot be

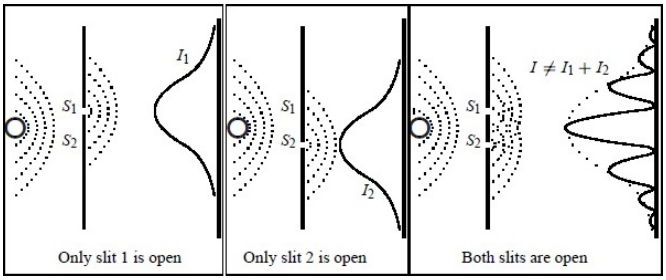


Figure 2.5: The Double-Slit Experiment: A source S emits light, which can pass through the slit S_1 and S_2, before striking the screen. I_1 and I_2 are the intensities recorded on the screen when S_1 is open, and then when only S_2 is open, respectively. When both slits are open, then the total intensity is no longer equal to the algebraic sum of intensities I_1 and I_2 (source)

explained in a deterministic way. An observer can only compute the probability $|\alpha_i|^2$ that the state vector ψ collapses to ϕ_i.

2.1.3.3 Density Operator

Consider some observable A which is in the *pure state* $|\psi\rangle$, and the expectation value expressed by:

$$\langle A \rangle_\psi = \langle \psi | A | \psi \rangle . \tag{2.8}$$

The definition is the following:

Definition 2.1.1. *The density matrix ρ of some pure state $|\psi\rangle$ can be defined as follows:*

$$\rho = |\psi\rangle \langle \psi| \tag{2.9}$$

The density matrix has the following properties:

- ρ is Projector : $\rho^2 = \rho$
- ρ is Hermitian: $\rho\dagger = \rho$
- positivity: $\rho \geq 0$
- Normalization: $Tr(\rho) = 1$

2.1.4 Signal Detection Theory

SDT [37, 10, 75] evolved with the development of radar and communication equipment. Initially, it was used in psychology (perception and sensation) in the early 50s and 60s to understand the human behavior feature when there was a presence of very complex signals that could not be described by the traditional theories [95, 96, 94]. Later, it was used for medical diagnostics, statistical decisions, etc. Almost all decision making and reasoning happen in the presence of uncertainty, and that is where SDT comes into play. SDT provides accurate graphic and language notation for estimating decision making in the presence of uncertainty. It is often used to analyze data obtained from some experiments where the task is to categorize the complex signal, which can be acquired by chance (known as noise in the SDT) or originated by some known process (known as signal in the SDT). For instance, radar operators must determine whether it is some parasite's presence (noise) or a plane's presence (signal) by observing the radar screen. This was the main application of SDT framework [37].

However, the context of signal and noise may be analogous in some experimental scenarios. For instance, in a face-memory experiment [10, 94, 113], each participant was provided a list of faces and asked to memorize them in the first part of the experiment. In the test phase, a set of faces was presented to the participants one at a time; the participants recognized some faces (old faces) and did not recognize others (new faces). In the experiment's initial stage, the main task was for the participants to identify whether they had seen each face before or not, by responding "Yes" if they had or "No" if they hadn't. In this case, the signal is related to the feelings of familiarity associated with memorized faces. The noise is related to the feelings associated to new faces. The general framework of SDT is shown in Figure 3.1, where the source, channel and receiver are present.

What are the different responses in this case? A "Yes" response for the given old faces is the right response, known as *Detection*; but a "Yes" response for the given new faces is a mistake, known as *False Alarm*. A "No" response for the given new faces is the right response, known as *Correct rejection*; but a "No" response for the given old faces is a mistake, known as *Miss*. The *Detection* and *Correct rejection* are good, but *False Alarm* and *Miss* are bad. These are the four different types of responses in SDT (see also Table 2.1). SDT supposes that internal response (internal response indicates internal impression or state of mind) varies randomly over each trail nearby the average value, generating a normal distribution of internal response curves (see Figure 2.6).

There are two essential components in the decision-making process: the decision criterion and the strength of the signal. In the case of signal strength, the Probability Density Function (PDF) is affected by the strength of the signal that leads of the shift in the signal+noise curve to the right (see Figure 2.6) if a stronger signal is present. The second component is the criterion in the decision-making process, which is quite different. The subject utilizes their own judgment in the decision-making process. Different subjects may decide that different error types are not similar. It is important to stress that subjects could be offered more money for getting more hits, but they could be cautioned not to make false alarms. The easiest and most effective strategy for the subject is to choose the location of *criterion* along the internal response axis (see Figure 2.6). The response is "Yes" whenever this criterion is less than the internal response, and the response is "No" whenever this criterion is higher than the internal response. In Figure 2.6, the vertical lines indicates the *criterion* response. These *criterion* response lines separate the graph into four different parts, specifically detect or hit, false alarm, miss, and correctly reject, which is also shown in Table 2.1. The internal response is higher than this *criterion* in the case of *False Alarm* and *Detection* when the response of the subject is "Yes". *Detection* is mainly related to the $signal + noise$ trials, as shown in Figure 2.6, when the internal response is higher than the criterion. Likewise, *False Alarm* is related to the $noise$ trials alone, as shown in Figure 2.6, when the internal response is higher than the criterion.

	Presence of signal (1)	Absence of signal (0)
Decision as Yes (1)	Detection (K_{11})	False Alarm (K_{01})
Decision as No (0)	Miss (K_{10})	Rejection (K_{00})

Table 2.1: Summary of decision about the presence or absence of signal. For each actual state and decision, the corresponding outcome has costs.

SDT deals with the choices regarding the hypothesis about the system at hand. There are two hypotheses in the binary decision scenario. Both hypotheses are illustrated by the presence or absence of a signal a in the input form x to the receiver during some observation interval $(0, T)$, where n is noise with some statistical characteristics. The hypotheses are written as follows:

- Null hypothesis, $A_0 : x = n$
- Alternative hypothesis, $A_1 : x = a + n$

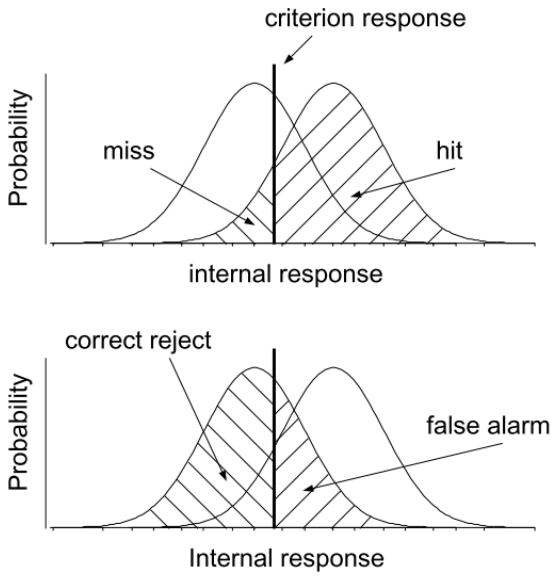


Figure 2.6: Distribution of internal response (internal response indicates an internal impression or state of mind) curve in SDT for noise and signal+noise trials.

There are two known hypothesis for PDF $p_0(x_1, x_2, \ldots, x_n)$ and $p_1(x_1, x_2, \ldots, x_n)$ of these data. There are two ways to choose between the two hypotheses.

If Bayesian decision theory is considered, so observer know:

- Hypothesis A_0 with the prior probabilities ξ, and hypothesis A_1 with the prior probabilities $(1 - \xi)$
- the four given costs K_{ij} of opting A_i when A_j is correct $(i, j \in \{0, 1\})$.

The costs are needed by the actions and circumstances following the decisions, so the average cost can be minimized. The decision costs are mentioned in Table 2.1. This is known as "Bayes strategy", which needs A_1 to be chosen when [48, 45, 47, 46]

$$\Lambda(x_1, x_2, \ldots, x_n) = \frac{p_1(x_1, x_2, \ldots, x_n)}{p_0(x_1, x_2, \ldots, x_n)} \geq \frac{\xi(K_{10} - K_{00})}{(1 - \xi)(K_{01} - K_{11})} = \Lambda_0 \qquad (2.10)$$

otherwise, A_0 is chosen. $\Lambda(x_1, x_2, \ldots, x_n)$ is known as the likelihood ratio.

The optimal binary choice can be interpreted in another way using the theory by Neyman and Pearson [79, 80]. Two types of error may occur:

- selecting A_1 when A_0 is correct, that is known as the first type of error or false alarm, and their probability under the provided approach is represented by Q_0;

- selecting A_0 when A_1 is correct, which is known as the second type of error or miss, and their probability is represented by Q_1.

The complement $Q_d = 1-Q_1$ is called a probability of detection. Identifying the optimal choice achieves the maximum probability of Q_d for every first type of error probability Q_0. These takes to the same Eq.2.10 of likelihood ratio $\Lambda(x_1, x_2, \ldots, x_n)$ with the given decision level Λ_0, but the K_{01} probability becomes equal to the preassigned value [48, 45, 47, 46] by fixing Λ_0.

Estimation theory generally manages data $x = (x_1, x_2, \ldots, x_n)$ whose joint PDF $P(x;\theta) = P(x_1, x_2, \ldots, x_n; \theta_1, \theta_2, \ldots, \theta_m)$, which is to be estimated depending on some unknown parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$. For instance, the data could be samples of some given input $x = a(\theta) + n$ to the receiver, including noise n with some statistical attribute and signal $a(\theta)$ based on parameter θ, i.e. amplitude, carrier frequency, and time of arrival.

The estimation of seriousness or cost of errors in the estimates $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m)$ of the parameters is set up by estimation theory. The cost function can be explained as a weighted sum of the squared errors,

$$K(\hat{\theta}, \theta) = \sum_{k=1}^{m} W_k(\hat{\theta}_k - \theta_k)^2. \tag{2.11}$$

The challenge is to identify the $\hat{\theta}_k = \hat{\theta}_k(x_1, x_2, x_3, \ldots, x_n)$ as a function of that data in order to minimize the average cost.

2.2 Literature Survey

The main focus of this survey is the effectiveness of classification approaches. There are numerous works about classification effectiveness and efficiency, but it is essential to limit the discussion. There is not enough work that uses quantum-inspired models for classification. This work uses decision theory, SDT, QM, quantum SDT, and classical ML approaches to propose quantum detection models for classification. SDT is the fundamental notion used often in this work, and there are several works where SDT is used for classification. So, this literature survey focuses on SDT and moves to the different ML approaches to improve classification effectiveness.

2.2.1 Signal Detection Theory for classification

SDT has also been used for classification. Several works discussed later in this thesis have found SDT to be effective in terms of classification images as well as when used with ML approaches.

2.2.1.1 Classification Images with Signal Detection Theory

Several studies discuss SDT with classification images. Abbey et al. [1] studied classification images acquired from the Two-Alternative Forced-Choice (2AFC) experimental model for the statistical and estimations hypothesis testing. Simple FC (forced-choice) discrimination and detection tasks are carried out using the probabilistic framework for task performance. The traditional linear filter frameworks are the main focus because a straight explanation of the classification images is obtained using these frameworks by providing an approximation of filter weights. Furthermore, the estimation process was explained for acquiring the classification images using observer data. To test different hypotheses using classification images based on some compact features set, several statistical tests are provided. To provide the details regarding how the proposed method can be used, a case study is carried out for the investigation of a Gaussian bump profile.

Murray et al. [75] studied the advancement in the classification images over recent years. The study provided the thorough review of classification images and provided the development of the frameworks using mathematical and statistical methods. It also described using how these frameworks helped to study some biological systems. An observer's responses are generally modeled on the basis of decision variables in SDT by correlating the template and signal, probably after corruption by internal and external noise. Estimating the effect of each

pixel of external noise on the responses to the observer, an observer's template is computed by using the response classification approach. A map used to describe each pixel of the image is known as a classification image. Some studies have computed classification images from the external noise field, but the optimal computation has never been formed, and the obtained quality of the classification images has never been computed. The optimal weighted sum of the noise field is derived for computing classification images in numerous experimental setups. The signal-to-noise ratio (SNR) is also derived from the resulting classification images [77]. It is demonstrated how to select the experimental parameters by utilizing the expression of SNR, i.e., the external noise power and the observer's performance level, to find the classification images with maximum SNR. In the 2AFC experiment, the signal is presented twice in order to compute the strength of the internal noise from the uniformity of responses from observers, where an observer rates the confidence interval of her/his response.

SDT is based on classical decision theory for perceptual decision making. SDT explains that when individuals make decisions based on audio or visual information, they compute some decision variable that encodes their approximation of the probability of every possible outcome being right. These decision variables are internal variables where task-relevant information is encoded. Nevertheless, the decision variable is generally contemplated to be a theoretical construct that cannot be computed directly. So this makes it very challenging to test the theories for perceptual decision making.

No work describes estimating those decision variables behaviorally. Pritchett et al. [83] proposed an approach to estimate those decision variables. The proposed approach is very effective in showing the decision strategies made by individuals. A new framework was investigated to resolve the long-standing problems in decision making and perceptions. The first process is to reveal the estimation of decision variables based on individual trials. These estimations will have been used to directly test the perceptual decision making theories. The measurement of classification images allows to obtain the estimates of templates used by the observer to extract information from the signal. To compute the observer's decision variables, the dot product of these signals with classification images is computed. Lastly, the decision space of each observer is constructed, allowing to reveal observer's responses, the probability of which is based on each value of the decision variables. The proposed approach has been applied in 2AFC tasks in order to investigate decision strategies. An obtained decision space found in one experiment assists the difference model, which is just the traditional frameworks of 2AFC decisions. An unexpected decision space was found in the second experiment, which cannot be predicted by using traditional frameworks of 2AFC decisions.

Some new evidence was found from these experiments about the observers' strategies in 2AFC.

2.2.1.2 Integration of Signal Detection Theory for Machine Learning classification tasks

The application of SDT is widely integrated with ML classification and approaches to solve several problems in different domains. For example, Gao et al. [36] investigated the classification of distinct metallic land-mine-like targets through SDT using EMI sensors. The approach assimilates the theoretical framework of the sensor response and uncertainties about the orientation of the sensor. This method allows the framework to be trained with less data collection, and the framework is evaluated utilizing experimental data. The framework showed effectiveness and provided better results than the baselines.

Michalopoulou et al. [71] trained multilayer perceptron using a backpropagation approach, which is experimented in the detection and classification task, and also compared with other approaches derived from the likelihood ratio tests. Several other works can also be found where the integration of SDT and classification is used for some tasks [55, 42].

2.2.2 Effectiveness of Classifier by using Different Machine Learning approaches

The performance of classification models heavily depends upon the complexity of data. The effectiveness of the classifier depends upon the diverse range of features, training samples, and categories. There are several works [4, 117, 50, 25, 12, 104, 23, 125, 18, 84, 92, 27, 24] where the ineffectiveness of classification models can be seen. The following study describes the related works based on problem.

Reuters newswire dataset is one of the most widely used test collections for text classification tasks. However, this dataset is used in several subsets, and mainly three subgroups are used often. These subsets are the sets with the top 10 categories with the highest number of positing training instances. The set with the top 90 categories consists of at least one positive training and test sample, and the set with 115 categories is comprised of at least one positive training instance [27]. The study by Yang et al. [120] can also be seen as an extensive evaluation of several statistical methods for text classification on different versions of Reuters datasets. Several other datasets have a large number of categories and features, such as 20 Newsgroups, TDT2, MNIST, etc.

2.2.2.1 Importance of essential Features in text classification

Joachims et al. [52] tried to explore the behavior of SVM classifier on Reuters-21578 and ModeApte datasets. The study found a particular expression of learning with SVM and why SVM is suitable for the specific task. Precision/Recall was used to show the effectiveness of the model. Different feature ranges were selected, such as 500, 1000, 2000, 5000, and 10000, to check SVM's behavior change. The study acknowledges the specific features that affect the SVM performance. Another work by Joachims et al. [51] used *TfIdf* with the Rocchio algorithm to improve the text classification performance.

The features quality are crucial with classification models. Yang et al. [121] did a thorough study of feature selection for the text classification task by selecting a range of features. The idea was to give special attention to the aggressive reduction of dimensionality. The different approaches, such as choosing terms based on document frequency, mutual information, term strength, information gain, and χ^2 test, were assessed. The experimental results showed that χ^2 test and information gain are the most effective approaches for improving the classification performance. KNN classifier with information gain thresholding improved classification accuracy by eliminating up to 98% of unique terms. The performance of KNN with document frequency was almost identical to information gain. Strong correlations were identified for the terms values between χ^2, information gain, and Document Frequency (DF). The experimental results suggested that DF thresholding may be a better choice due to its simplicity and lower computational cost, compared to using χ^2 or information gain, which are computationally expensive.

However, some widely known feature selection approaches like χ^2 or information gain have a greedy nature and cannot always be very useful due to specific criteria. If the reserved data dimensionality is very low, it can lead to the poor performance of these greedy nature approaches. Yang et al. [118] proposed a better feature selection approach for improving text classification performance. The discrete space's proposed approach optimizes the orthogonal centroid subspace learning model's objective function, known as orthogonal centroid feature selection. The proposed method was applied to several datasets, and the obtained results showed effectiveness on several datasets.

There are so many noisy features that automatically reduce the performance of any classifiers. Another work by Yang et al. [119] focused on removing those noisy data by using Linear Least-Square Fit (LLSF) mapping. Several noise reduction approaches were proposed

and evaluated: removing non-informative words before the training step, cutting off noisy latent semantic structure by using singular value decomposition during the training step, and eliminating non-influential components in LLSF after the training step. The proposed study showed effectiveness in classification accuracy on several datasets. Koller et al. [57] examined a feature selection approach for selecting essential features based on information theory for improving classification performance. The proposed feature selection approach used cross-entropy to reduce the necessary information lost during the feature elimination step. It showed that there are several desirable properties for the feature selection procedure from the theoretical framework. Furthermore, an algorithm was presented to approximate the theoretical approach and provide large-scale experimental testing. The algorithm has shown to be competent in improving classification performance by reducing the feature space dimension and improving learning tasks.

2.2.2.2 Focus on Misfit Problem to Improve the Effectiveness

Generally, when the data fit the model, accuracy is high, but when they does not fit, accuracy is low. Wu et al. [114] proposed a refinement model to deal with such misfit problems. The proposed works suggested that it is not required to change the classification approach to improve accuracy. Instead, successive refinements can be used to classify training data to correct this model's misfit problem. The proposed approach was applied to the Rocchio and Naive Bayes classifier to enhance classifier performance. The proposed approach seemed to be very useful with the large text collections since data can be stored on disk, and only one scan of data is required to build classifiers. The proposed model was applied on two dataset, and obtained results showed a 45% improvement in effectiveness on average. Han et al. [39] proposed a document classification model based on centroid. This model is straightforward as well as robust. The experimental results showed that this approach is very effective and outperformed baselines, i.e., KNN, NB, and DT, on several datasets. The experimental results demonstrated that the centroid-based approach's similarity measure allows for the classification of new document samples. It is based on how their behavior nearly matches document samples' behavior corresponding to the distinct classes. The similarity function allows for the dynamical adjustment for classes with different densities and it is responsible for the dependencies among the terms in the distinct classes. Tan et al. [97] proposed the DragPushing model for the text classification task. The proposed model computes centroids for every document class by using the training set. Furthermore, the model iteratively filters these centroids using misclassified instances by pulling the correct centroids towards the misclassified

instances. At the same time, centroids of an incorrect class are pushed away from the misclassified instances. The proposed approach is computationally efficient and easy to implement. The obtained results on datasets have shown effectiveness to some extent.

2.2.2.3 Classification Effectiveness followed by a Clustering Approach

Kyriakopoulou et al. [59] tackled an issue of learning by taking advantage of information obtained from the clustering of both testing and training sets for text classification. The classification performance is expected to improve by integrating information obtained from clustering into the feature space representation of texts. The training datasets were used in different sizes including 0.5%, 1%, 5% and up to 100% of the training set. The proposed SVM model with clustering performed very well on all datasets by providing, on average, a 15% precision/recall breakeven point improvement. Zeng et al. [126] proposed a text classifier based on the clustering approach. The proposed approach is called a semi-supervised learning approach because it uses labelled and unlabelled training samples during the training stage. The accuracy of existing models was not satisfactory due to the data distribution and small size of labelled data, but unlabelled data samples were able to enhance the performance of the trained model to some extent. The Clustering-based Classification (CBC) technique was proposed to solve this issue. Both labelled and unlabelled training data were clustered using the CBC technique with labelled data. Furthermore, several unlabelled samples were labelled using the obtained clusters. The CBC model has shown effectiveness by outperforming several baselines when the labelled dataset size was minimal.

2.2.2.4 To improve the classification performance using Quantum-Inspired approaches

Some recent works use quantum-like approaches to improve classifier performance. The kernel method is generally used with SVM to improve classification performance, but if the noise label is high in the dataset, it's hard to achieve good performance with existing kernels. Tiwari et al. [98] proposed a novel quantum kernel to improve the SVM classifier performance by dealing with the data non-separability issue. The experiment was performed on two toy datasets, which are often used to check the performance of kernel methods. It was obtained from the experimental analysis that the proposed kernel improves the classification performance by providing a cleaner decision boundary. The proposed kernel was tested on several parameters, and it proved to be effective at a high noise ratio where other models were

not.

Han et al. [40] proposed the Binary Gravitational Search Algorithm (BGSA)-KNN model, which seeks to remove the irrelevant feature space and improve the classification accuracy by incorporating the quantum-inspired BGSA [85] and 1-nearest neighbor approach estimated by the Leave-One-Out Cross-Validation (LOCV) [89]. The main aim was to optimize the feature selection subset effectively. The proposed model was tested on several datasets and achieved higher accuracy than baselines. The success of the quantum-inspired BGSA was due to the merge of BGSA and QM, avoiding the unreasonable convergence of the two and enhancing the potential of exploitation and exploration. Hence, the quantum-inspired BGSA effectively optimized the feature selection set.

Nasios et al. [78] proposed a non-parametric evaluation model inspired by QM. Kernel density evaluation relates a function to every data instance. The density function is computed by summing all the kernels in classical estimation theory. The proposed model supposes that each data instance is incorporated into the quantum particles with radial activation. The Schrodinger equation was used to explain the position of particles by observing energy levels. Every data instance's position was known, and their associated PDF was modelled utilizing correspondence with the quantum function. The distribution of KNN statistics was used to compute the kernel scale. The local Hessian was used to identify the modes in the potential quantum hypersurface for pattern classification. Every mode was incorporated into some nonparametric class that was explained utilizing a region growing algorithm. The proposed model was implemented for the topography segmentation using the satellite image data and artificial data. Zhang et al. [127] proposed complex-valued fuzzy network (CFN) for the sarcasm detection task by integrating the idea of quantum mechanics and fuzzy logic.

2.2.2.5 Kernel methods Inspired by Quantum Theory for Classification

Few recent works use quantum-inspired kernels to improve classification effectiveness. Generally, real-world datasets are non linearly separable. In the classification tasks, SVM uses a kernel function to solve the non-linearity problem, which transforms the non linearly separable data into a high dimensional space where it becomes separable. Several existing kernel functions are used with SVM, and one of the most famous is the Radial Bias Function (RBF) kernel. However, there are still some shortcoming with the RBF kernel. It cannot provide a very clear decision boundary when non-linearity is very high, so it becomes difficult for the classifier to obtain high performance. When the feature space becomes vast, the existing kernel is computationally expensive while training and prediction for the new data points.

Tiwari et al. [98] proposed a novel quantum kernel in order to improve the SVM classifier performance. The proposed kernel's main feature is its ability to deal with a high Gaussian noise ratio, whereas others cannot. The experimental results have shown that it provides a cleaner decision boundary than other kernels.

Chatterjee et al. [19] proposed nonlinear kernel functions based on the generalized and canonical coherent states. The primary connection was the property of the Reproducing Kernel Hilbert Space (RKHS) with SVM that naturally appears from the generalized and canonical coherent states. A Positive Operator-Valued Measure (POVM) was used for the efficient computation of radial kernels on the quantum system, which is based on a canonical coherent state.

Schuld et al. [91] proposed a kernel to handle non linearly separable data and applied the kernel to SVM. The main idea was to encode the inputs as a nonlinear feature map into a quantum state that transforms those data into a quantum Hilbert space. To check the kernel's performance with SVM, 2-dimensional benchmark datasets were used, and the obtained results were compared with RBF. The proposed kernel performed better than RBF and provided a cleaner decision boundary than RBF.

Havlivcek et al. [41] proposed two quantum kernel algorithms and applied them to SVM. The main aim of both approaches was to use a quantum state as a feature space. These proposed models have shown to be effective when applied to SVM for the classification problem.

2.2.2.6 Traditional Kernel Methods to Improve Classification Performance

Several existing kernels use classical theory to deal with the non-linearity. The kernels have been used very widely with SVM and with other ML algorithms. The kernel-based SVM resulted in a reasonable performance, but these approaches in online settings are less applicable for real-time applications and have not been explored much. Kivinen et al. [56] studied online learning in the RKHS space. The proposed method was computationally powerful using stochastic gradient descent within a given feature space and some other tricks. The value of a large margin was exploited for the classification task in the online settings. The model proved to be effective for the classification task.

Rosipal et al. [87] proposed a novel kernel method for classification known as Partial Least Squares (PLS)-Support Vector Classifier (SVC). The proposed framework was based on the PLS followed by an SVC. Generally, Principal Component Analysis (PCA) was used as a dimensionality reduction approach for discrimination problems. In contrast, orthonormalized PLS is very close to Fisher's technique [32] in correlation analysis or linear discrimina-

tions. It is the reason that orthonormalized PLS may be more suitable for PCA for discrimination. The PLS-SVC model was applied to 13 different benchmark datasets. The obtained results were shown to be effective and outperformed state-of-the-art (SOTA) classification models.

Chen et al. [20] proposed a subspace kernel based on the Hilbert-Schmidt Independence Criterion (HSIC). It achieved an optimal subspace kernel by resolving the eigenvalue issue. The main limitation with the existing kernel was that the learning kernel and the classification are independent. The join optimization approach was proposed to learn the subspace kernel and classification simultaneously to resolve such limitations. Furthermore, the learning formulation was proposed to eliminate the noisy information in a subspace kernel by extracting uncorrelated subspace kernels. The proposed formulation was extended to multiple kernel integration when multiple kernels are available. The experimental results showed the effectiveness of the proposed model on several benchmark datasets. A linear classifier fails to perform when it comes to complex classification problems, which leads to a non-linear classifier. Fung et al. [35] tried to use a non-linear kernel classifier on large datasets. The proposed kernel classifier used a leave-one-out error bound based on the linear program, which constructs a nonlinear kernel classifier that mainly depends on 10% of the data of the original set. It led to reduced kernel data dependency, which allowed for fast classification due to fewer storage requirements.

Fung et al. [33] proposed a nonlinear kernel for SVM classifier. Prior knowledge was used in the shape of multiple polyhedral sets, each corresponding to one of two classes. A linear programming formulation with a nonlinear symmetric kernel was used to obtain the classifier. The proposed classifier was the extended version [34], which uses the same prior information for linear SVM.

Eskin et al. [30] proposed a mismatch kernel, which is a class of string kernels. The proposed approach was used with SVM in a discriminative method for the protein classification task. Sequence similarity was computed by the kernel-based on shared occurrences of some k length subsequences, enumerated up to some m mismatches, and without depending upon positive training sequences. Kernels were computed very effectively by using a mismatch tree data structure. The proposed approach was applied to a few benchmarks datasets, and it proved to be effective with SVM classification tasks. Some reviews papers about learning with kernels can be found in these works [90, 49].

2.2.2.7 Neural Networks based models for classification

In recent years, NN has been used extensively for text classification tasks. Jiang et al. [50] proposed a hybrid text classification approach by using softmax regression and deep belief networks. A deep belief network was applied to tackle the computation problem of a sparse high dimensional matrix of text data. Furthermore, softmax regression was adopted in the learned feature space to classify the text. Firstly, softmax regression with a deep belief network was trained in the pre-training procedure. The obtained output was mapped into coherent of the whole, and the Limited-memory Broyden–Fletcher–Goldfarb–Shannon (BFGS) was used to optimize the system parameters in the fine-tuning stage [62]. The experiment was performed on 20-Newsgroup and Reuters-21578, consisting of 20 classes in 20-Newsgroup and 135 classes in Reuters-21578 with large feature spaces. The proposed model improved accuracy on the range of training and testing data and compared it with SVM and KNN. The existing back propagation neural network approaches prolonged speed in the training process and had a chance to stick into local minima, which led to poor performance and low efficiency. Wang et al. [111] proposed a text classification model by using Latent Semantic Analysis (LSA) and Modified Backpropagation Neural Network (MBPNN), which improve the performance and increase training speed. The integration of LSA allows to solve the issues raised due to the statistically derived conceptual indices by replacing individual words, which also assists in the drastic reduction of dimensions. The use of MBPNN and LSA improved the text classification task's efficiency and accuracy. Another similar work can be found here [124]. Zheng et al. [129] proposed a text classification model based on the Regularization Extreme Learning Machine (RELM) with the integration of LSA. An experimental analysis was performed on the range of features on Reuters-21578 (top 10 categories) and WebKB datasets. The obtained results proved to be effective and the model learned faster than other models like SVM and feed-forward neural networks.

CHAPTER 3

Methodology

3.1 Introduction

This section starts with the proposed approach to handle the classification task. The proposed *QDM classifier* * [102, 101, 100, 28] uses a quantum-inspired paradigm instead of a classical paradigm. Quantum-inspired classifier deals with the ineffectiveness that arises from a diverse range of features, training samples and categories, and inflexibility in hyperparamater tuning (RQ1, RQ2, RQ3, RQ4). In previous chapters, the details about the quantum detection model are described in Section 3.1.1. Furthermore, the proposed framework can be found in Section 3.2.3.1 along with the pseudocode in Section 3.2.3.2. This chapter also refers to the published works [102, 101, 100, 28] of the Ph.D. candidate. The proposed approach is a binary classifier. Generally, multi-class and multi-label classification problems are also solved using a binary classifier by decomposing them into binary classification problem [26, 3, 7] in traditional ML tasks.

3.1.1 Quantum Signal Detection Theory

The main purpose of applying SDT to classification is to provide structured strategies for constructing frameworks that allow to reduce the average number of decision errors. The main goal of quantum SDT is to find how the quantum mechanical uncertainty and random noise affect the stability of the decisions and parameter estimates [16].

The main idea behind detection is to find the hidden information in the data transmitted

*https://github.com/prayagtiwari/QDM

by the source to the detector. The data sent through the channel is only an **approximation** of the *right* information that the source wants to send. For instance, an IR system sends relevant information to the user solely based on user's information needs, which means the document is primarily an approximation of the information satisfying the user's needs.

The standard communication framework consists of a source, channel, and receiver (see Figure 3.1). The source sends a signal a (*right* information) selected from some fixed alphabets. For instance, alphabets could be binary, and the transmitted signal would be either 0 or 1. The signal a is transmitted through a channel to the receiver, and the symbol x is measured by the receiver. The measured signal x could be different from the original signal a due to the presence of noise.

Like classical communication frameworks (see Figure 3.1), the quantum communication frameworks (see Figure 3.2) also has a source, a channel, and a receiver. Classical information is transmitted by a quantum channel using quantum states as the information carriers. This system is known as the quantum communication framework. On one side, the signal is encoded into a particle (e.g., a qubit) by the coder, and then a pure state is assigned to the particle described by its pure state vector $|\phi\rangle$. Finally, a measurement is conducted as in a classical detection system, on the other side [17].

A coder plays an important role between the source and the channel in the quantum SDT as demonstrated in Figure 3.2. The distinction between the classical and quantum systems can be explained by what the encoder encodes and what the decoder decodes [16]. Mathematically speaking, this contrast means that the projector corresponding to the prime measurement can optimally be measured by utilizing classical theory in the case of the classical model [47, 46, 45].

In the classical communication framework, initial encoding takes place at the encoder side with classical to classical (c-c) mapping from symbol to wave, which is transmitted to the corrupted channel. Classical to classical (c-c) mapping also occurs at the decoder side, where the corrupted wave is decoded to the symbol. So, the overall operation is performed in classical settings in the classical framework.

In contrast, classical information, or symbol, is transmitted by a quantum state in the quantum communication frameworks. The initial encoding occurs at the encoder side with classical to quantum (c-q) mapping from symbol to quantum state, which is chosen from some set of finite states. Furthermore, the quantum channel provides quantum to quantum (q-q) mapping from quantum states to corrupted versions. Quantum to classical (q-c) mapping takes place at the decoder side, and a measurement is conducted in the same way as it is

in a classical system.

Generally speaking, the potential of a quantum system is the degrees of freedom in setting the decoder, which may be set to measure distinct observables. On the one hand, classical symbols are selected from a finite set of candidates (e.g., feature vocabulary). On the other hand, this c-q and q-c mappings could help obtain more effective signals as these signals are less susceptible to classification errors.

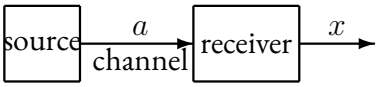


Figure 3.1: Classical communication system

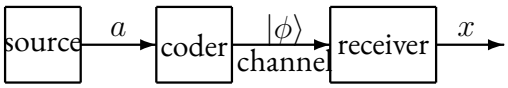


Figure 3.2: Quantum communication system

Mathematically, vectors, projectors, and density matrices are used to represent signals, states, and observables. The basis vectors belong to some observable values, and the state of the signal is represented by state vectors. The probability of receiving symbol x to the detector based on some state v_j is estimated by $|x'v_j|^2$. In the alphabet case comprised of binary values, the state vectors are represented by v_0 and v_1. As observables comprise values, it is essential to describe the elements to be observed to attain the observables, which are utilized to identify the particle states transmitted via the channel. The set of observables values takes part in order to make the decision about the particle state. For instance, in the case of a group of two states and binary observables value, the value of $x = 1$ can indicate v_1, and the value of $x = 0$ can indicate v_0.

The probability $p(v_i|v_j)$ that the final decision is v_i given state v_j is estimated after splitting the group of observables values. In terms of representation, the channel could be the probability network allowing us to estimate the degree of disturbance of an emitted signal during the channel measurement, and symbol x is observed, allowing us to decide whether the emitted symbol was a_1 or a_0. The optimal decision v_i goes by x, which is finally generated in turn into v_j with some probability $|x'v_j|^2$.

The detection task comprises expressing the group of observable values belonging to the states in which signal can be transmitted via the channel. This allows to maximize the probability of correct detection and reduce the probability of error.

3.2 Quantum Detection Model

This section proposes a classification model to deal with the problem of effectiveness and efficiency. The proposed model has a high degree of flexibility in terms of hyperparameter tuning. The details about SDT [2, 37] can be found in Section 2.1.4, and quantum SDT [48, 45, 47, 46] in Section 3.1.1. The code of the proposed model along with the published papers [102, 101, 100, 28] can be found on Github * as well. The pseudocode of the proposed QDM model can be found in Section 3.2.3.2.

3.2.1 The Detection Operator

In the two given hypotheses regarding the decision, they simply mean absence of and presence of a signal. Generally, the decisions regarding the system need to be taken between two given density operators in the case of a quantum scenario. The two density operators are either:

- A_0 with prior probability ξ and density operator ρ_0, or
- A_1 with prior probability $(1-\xi)$ and density operator ρ_1

Also the four given costs K_{ij} of opting hypotheses A_i when A_j is correct $(i, j \in \{0, 1\})$. Consider output $x_1, x_2, x_3, \dots$ of some commuting observables $X_1, X_2, X_3, \dots$. The decision is based upon the resultant of some function $f(x_1, x_2, x_3, \dots)$. Likewise, this can be based on the result of a measurement of the given operator $f(X_1, X_2, X_3, \dots)$. So, the question is, how should the operator be defined ?

The main requirement here is the result must be one of two numbers (0, and 1). A_1 must be chosen if it is 1 and A_0 must be chosen if it is 0. The operator $f(X_1, X_2, X_3, \dots)$ is the one whose whose eigenvalues must be 0 and 1. We call such an operator a *projection operator (detection operator)* and it is denoted by Δ. The main question is, Which is the best detection operator Δ for the system? We simply use the formulation of the average cost to minimize this cost for all detection operator sets to find the best Δ. Q_0 is the probabilities of error of the first type (also known as a false alarm), and it happens due to choosing A_1 when A_0 is correct. Q_1 is the probabilities of error of the second type or miss. The complement $Q_d = 1 - Q_1$ is called a probability of detection. The average cost depends upon the probabilities error Q_0 and Q_1. The measurement of detection operator gives the value 1 under A_0 when A_1 is chosen.

*https://github.com/prayagtiwari/QDM

$$Q_0 = Pr(\Delta \to 1|A_0) = E(\Delta|A_0) = \text{Tr}(\rho_0\Delta) \tag{3.1}$$

Where, E denotes the expected value.

Similarly,

$$Q_1 = Tr[\rho_1(1-\Delta)] = 1 - \text{Tr}(\rho_1\Delta) \tag{3.2}$$

So, the average cost is as follows:

$$\begin{aligned} \bar{K} &= \xi[K_{00}(1-Q_0) + K_{01}Q_0] + (1-\xi)[K_{01}Q_1 + K_{11}(1-Q_1)] \\ &= \xi K_{00} + (1-\xi)K_{01} - (1-\xi)(K_{01} - K_{11})\text{Tr}(\rho_1 - \lambda\rho_0)\Delta \end{aligned} \tag{3.3}$$

where

$$\lambda = \frac{\xi(K_{10} - K_{00})}{(1-\xi)(K_{01} - K_{11})} \tag{3.4}$$

As long as $K_{01} > K_{11}$, $\bar{K}$ can only be minimum if $Tr((\rho_1 - \lambda\rho_0)\Delta)$ is maximized.

The optimal projection operator is obtained by the eigenstates $|e_j\rangle$ of the operator $(\rho_1 - \lambda\rho_0)$ refers to the positive eigenvalues, so the eigensystem can be written by

$$(\rho_1 - \lambda\rho_0)\,|e_j\rangle = e_j\,|e_j\rangle \qquad j = 1, \ldots, \text{rank of } \rho_1 - \lambda\rho_0 \tag{3.5}$$

so the requirement to maximise can be re-written as

$$\text{Tr}(\rho_1 - \lambda\rho_0)\Delta = \sum_j e_j \langle e_j|\Delta|e_j\rangle \tag{3.6}$$

and this can be obtained if

$$e_j \langle e_j|\Delta|e_j\rangle = 1, e_j \geq 0 \quad \text{and} \quad e_j \langle e_j|\Delta|e_j\rangle = 0, e_j < 0.$$

So, the estimation of the optimal detection operator between A_0 and A_1 is as follows:

$$\Delta = \sum_{j:e_j \geq 0} |e_j\rangle \langle e_j| \tag{3.7}$$

So the probabilities of error are written as follows:

$$Q_0 = \sum_{j:e_j \geq 0} \langle e_j|\rho_0|e_j\rangle\,, \qquad Q_1 = 1 - \sum_{j:e_j \geq 0} \langle e_j|\rho_1|e_j\rangle \tag{3.8}$$

and the minimum average cost can be written as

$$\bar{K}_{min} = \xi K_{00} + (1-\xi)K_{01} - (1-\xi)(K_{01} - K_{11}) \sum_{j:v_j > 0} e_j \tag{3.9}$$

3.2.2 Quantum Signal Detection Theory in IR

It is worthwhile to mention that quantum signal detection theory (SDT) was used in IR [67] to re-weight the query terms and then re-rank the retrieved documents. The main aim was to replace the probabilistic relevance feedback framework based on BM25 and the vector space relevance feedback framework based on the vector space model by quantum SDT. Firstly, the notion of SDT was explained in terms of quantum probability theory. QM is the generalization of classical theory, and some quantum probability distributions cannot be explained within the traditional probability theory, which has led us to find the prime solutions. QM also uses matrices and vectors, which leads to the fusion of the proposed framework in the vector space model.

In general, query vector is formed as the optimal detector of quantum SDT. The optimal detectors need to decide the relevance state of some document based on the data (for example, term frequency of query). Technically, the original query vector is projected using these algorithms on a particular subspace provided by the idea of quantum SDT. The obtained vector from the projection is matched against the document vector by using $x'y$ (inner product function). More formally, the vector spanning the subspace is v_1, which is provided by the idea of quantum SDT.

Moreover, optimal detectors are described from the prepared setting with regard to the quantum probability distribution. The optimal detectors are said to be the eigenvectors, which is the particular type of matrix composed of the quantum probability distribution. The classical probability theory cannot identify these optimal detectors. Eventually, the query vectors are projected on the eigenvectors, which are identified by the quantum probability distribution.

Suppose an input query of the IR system is denoted by a vector $|y\rangle$, which is representing a ranked list of documents, and a vector $|x\rangle$ is denoting each of the documents. The relevance

assessments allow us to compute the density operator ρ_1 by utilizing relevant documents and ρ_0 by utilizing non-relevant documents, in order to obtain the eigenvectors. The projection of the original query vector is represented by the optimal detection operator Δ with query vector y. The quantum SDT is used for the relevance feedback algorithm in IR by projecting both the document vector $|x\rangle$ and the query vector $|y\rangle$ by means of the best detection operator; thus, re-ranking is estimated by

$$\langle x|\Delta|y\rangle.$$

3.2.3 Supervised Learning Framework with the Proposed Quantum Detection Model

The proposed framework is similar to the supervised ML, as shown in Figure 3.3. In the training process, the training data is transformed into a low dimension using feature selection methods. It is then fed to the QDM with the supervision of labels like other supervised ML models. In the prediction phase, unlabelled data is transformed into a low dimension and then fed to the trained quantum detection model where the prediction takes place.

The curse of dimensionality [105, 60] is a challenging problem in ML, and it means that the performance of ML models start decreasing with an increasing number of features. Theoretically speaking, high dimension allows more information to be stored, but in practice, this is not the case due to the presence of highly irrelevant features in real-world data. Generally speaking, when the dimension increases, the volume of space starts expanding, so the data becomes very sparse. The classifier performance improves with an increasing number of features until it reaches a certain range of top selected features. However, when the feature size increases without increasing training sample size, classifier performance starts deteriorating. It is essential to use the feature selection and cross-validation strategies in order to avoid the overfitting problem because of the problem of dimensionality. How many features should be used for the classification problem? There is no fixed rule for this. It mainly depends on the size of training samples, the type of classifier, and the complexity of decision boundaries.

Feature selection or dimensionality reduction strategies are used to avoid the dimensionality issue by selecting some top features. For feature selection, chi-square [38, 81] is used to select a range of top features. χ^2 is used widely in statistics to test the independence of two distinct events especially for the categorical attributes in a dataset. χ^2 is computed between each attribute and target, and a number of desired attributes with the best χ^2 scores are selected. Some features with the best χ^2 score values are visualized in Section 4.6.2 and Figure

4.8.

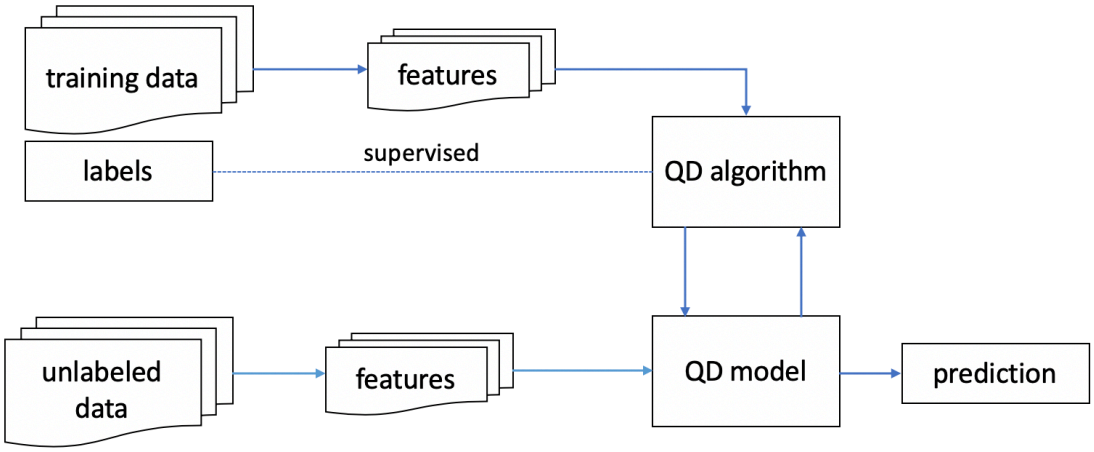


Figure 3.3: Proposed Quantum Detection Model (QDM) for Classification

3.2.3.1 Proposed Quantum Detection Model (QDM) for Classification

A novel classification model is proposed in this section, which is inspired by the quantum detection model.

Task Description. Like traditional supervised learning tasks, we also have features and corresponding labels as model inputs and outputs. Each training sample belongs to one of the multiple categories. Consider a collection of distinct features computed from the whole dataset. Every sample of the collection can be described as a feature vector, where every element is a non-negative number in the feature vector representation. Every training sample in the collection has a binary label $\{0, 1\}$.

Algorithm Overview. Our model makes a decision on whether a sample belongs to a category or not. Inspired by the quantum detection model, we estimate a projector Δ (refer to Section 3.1.1 for the detection operator Δ) from the training data to identify whether a sample belongs to the given category or not. In the test phase, it is computed against the vectorial representation for the given test sample to identify whether the test sample belongs to the category or not.

Density Operator Estimation. Our model first computes the density operators, ρ_0 and ρ_1, by utilizing the training sample. Specifically, for every class, the positive training sample is used to compute ρ_1, and the negative training sample is used to calculate ρ_0. Firstly, the total number of a set with the non-zero value is calculated for every specific feature to get the

density operators ρ_0 and ρ_1. In this way, one vector $|v\rangle$ is attained for every class. So, two vectors $|v_0\rangle$ and $|v_1\rangle$ are attained for negative and positive samples (follow lines 6-13 and 14-21 in Algorithm 3.1 to compute $|v_1\rangle$ and $|v_0\rangle$). These obtained vectors can be considered statistics of the feature vector in a class. Here, $|v_0\rangle$ and $|v_1\rangle$ are normalized in order to obtain $|\langle v|v\rangle|^2 = 1$. Then, the outer product is computed here to get the density operator ρ_0 and ρ_1 as follows:

$$\rho_0 = |v_0\rangle \langle v_0|, \qquad \rho_1 = |v_1\rangle \langle v_1| \tag{3.10}$$

since v_0 and v_1 are unit vectors, $Tr(|v_0\rangle \langle v_0|) = \langle v_0| |v_0\rangle = 1$.

Projection Operator Calculation. The projection operator Δ is computed according to Section 3.1.1 and Eq. 3.7. Essentially, the optimal detection keeps all eigenstates with non-negative eigenvalues as positive class and negative eigenstates as negative class, i.e. $\Delta = \sum_{e_j>=0} |e_j\rangle \langle e_j|$, $\Delta^{\perp} = \sum_{e_j<0} |e_j\rangle \langle e_j|$, where ξ denotes a prior probability of the negative class and $\lambda = \xi \,/\, (1-\xi)$. In addition, e_j denotes the positive eigenvalue associated to Δ, which denotes the vector subspaces representing the samples to considered in the target class.

We keep $\lambda = 1$, which means that the prior probability is kept the same for both of the classes ($\xi = 0.5$); furthermore, there is no cost for correct rejection and detection $K_{00} = K_{11} = 0$. Eventually, the costs for the false and miss are constant ($K_{01} = K_{10}$).

Decision Making. In the test phase, we use the obtained projectors to make decisions for each test sample. Essentially, we encode a test sample x as a state $|x\rangle$ with the same approach for constructing v_0 and v_1. Finally, a binary label is computed for the given test sample x, represented by $|x\rangle$, by examining the value of $\langle x|\Delta|x\rangle$. If $\langle x|\Delta|x\rangle \geq 0.5$ then x is allocated to the positive class, otherwise it's negative.

3.2.3.2 Pseudocode and Data Representation

The proposed QDM is written in Matlab, and the pseudocode of QDM can be found in Algorithm 3.2.3.2. The notations used in this pseudocode are as follows: X is the training set, r is the training labels, and F is the test set in the Algorithm 3.2.3.2. Some Matlab commands are used in the Algorithm 3.2.3.2 , i.e., *size* (in line 2) returns a row vector consisting of elements which are the corresponding dimensions, *zeros* returns a matrix consists of zero, *sum* returns a summation of elements, and *sign* is a signum function.

In Algorithm 3.2.3.2 (lines 3-5), N represents the number of samples in the training set,

k represents the number of features, R is the total number of samples in the category, and r is the labels.

Generally, squared roots of the probabilities are the elements of state vector v. In the Algorithm 3.2.3.2 (lines 6-21), the state of the positive samples is represented by a state vector $v_1 = (\sqrt{p_1} \ldots \sqrt{p_k})$, and negative samples by $v_0 = (\sqrt{q_1} \ldots \sqrt{q_k})$,
whereas $p_j = \frac{\text{\#positive samples indexed by j} + 0.5}{\text{\#positive samples} + 1}$ and $q_j = \frac{\text{\#negative samples indexed by j} + 0.5}{\text{\#negative samples} + 1}$.

Furthermore, eigenvector $|e_j\rangle$ is computed by Eq.3.5, (lines 23-30), where the value of $\lambda = 1$ (in line 23 of Algorithm 3.2.3.2). $eigs$ command is used and it returns matrix sB containing the eigenvalues, and UB matrix columns are the corresponding eigenvectors. If the first element of sB is greater than 0, then $|e_1\rangle = eigB1$ is obtained by selecting the first column with all row elements, and $|e_0\rangle = eigB0$ is obtained by selecting the top $1 : (k-2)$ column (as shown in lines 24-26), otherwise opposite (as shown in lines 27-30).

In the prediction phase (lines 31-43), if $\langle x|\Delta|x\rangle$ (line 36) is greater than 0.5, then it belongs to the positive category, otherwise it is negative. The test sample $x = F$ is encoded as state $|x\rangle$ and obtained using the same approach for constructing v_0 and v_1 (see line 35). In this, Δ is used to match the test sample; either it belongs to the category or not. The projection operator Δ is computed according to Section 3.1.1 and Eq. 3.7 (see line 35).

Algorithm 3.1 Quantum Detection Model (QDM) for classification

procedure P = QDETECT(X, r, F)
    trSetDim ← size(X);
    N ← trSetDim(1); #number of samples in the training set
    k ← trSetDim(2); #number of features
    R ← sum(r); #R is total number of samples in category, r is labels
    $p \leftarrow zeros()$;
    for j = 1:k
        $p(j) \leftarrow 0$;
        $p(j) \leftarrow (sum(sign(X(1:N,j)).*(r(1:N)))+0.5)/(R+1)$;
    end
    if (sum(p)>0)
        $p \leftarrow \sqrt{(p/sum(p))}$;
    end
    $q \leftarrow zeros()$;
    for j = 1:k
        $q(j) \leftarrow 0$;
        $q(j) \leftarrow (sum(sign(X(1:N,j)).*(1-r(1:N)))+0.5)/(sum(1-r(1:N))+1)$;
    end
    if (sum(q)>0)
        $q \leftarrow \sqrt{(q/sum(q))}$;
    end
    if (k>2 && isreal(q) && isreal(p) && sum(isnan(p))==0 && sum(isnan(q))==0)
        $[UB, sB] \leftarrow eigs(p' * p - q' * q, k-2)$;
        if (sB(1)>0)
            $eigB1 \leftarrow UB(:,1)$;
            $eigB0 \leftarrow UB(:,1:k-2)$;
        else
            $eigB1 \leftarrow UB(:,1:k-2)$;
            $eigB0 \leftarrow UB(:,1)$;
        end
        $predictions \leftarrow zeros(size(F,1),1)$;
        $s1 \leftarrow zeros(size(F,1),1)$;
        $s0 \leftarrow zeros(size(F,1),1)$;
        for j=1:size(F,1)
            $s1(j,1) \leftarrow trace(F(j,:) * (eigB1 * eigB1') * F(j,:)')$;
            if (s1(j,1)>0.5)
                $predictions(j,1) \leftarrow 1$;
            end
        end
        $P \leftarrow predictions$;
    else
        $P \leftarrow NaN(size(F,1),1)$;
    end

CHAPTER 4

Experiments

4.1 Introduction

The main aim of this section is to provide a summary of the experimental result on the proposed QDM for classification with the focus on resolving issues raised due to the complexity of data. The main research objective is to focus on the effectiveness resulting from a diverse range of features, training samples and categories. Furthermore, another focus of our research is the efficiency along with the performance of classification models. Well known text and image classification datasets like Reuters21578, 20 Newsgroup, MNIST handwritten image dataset, and TDT2 (Nist Topic Detection and Tracking corpus) are used for evaluating the effectiveness of QDM with several baselines. The description of the datasets can be found in Section 4.2. The experimental analysis is done thoroughly on the feature label and a diverse range of training samples for a number of categories in order to see the effectiveness of QDM. Furthermore, efficiency analysis is also performed for the QDM with baselines on a range of features as well as diverse training samples.

In this section, the experiment carried out on the proposed QDM for classification is explained. The proposed model name, QDM [100, 102, 101, 28] is used in this thesis. The code of the proposed model along with the published papers can be found on Github *.

*https://github.com/prayagtiwari/QDM

4.2 Dataset Description

4.2.1 Reuters21578

Reuters21578 [*] is one of the most broadly used test collections for text classification tasks. This data was initially collected by the Reuters, Ltd and Carnegie Group, which was published in 1987 in the Reuters newswires as a test collection. There are 21,578 documents contained in this corpus across 135 categories. Several documents with multi-label categories have been removed from the corpus, leaving it with 65 categories. There were 18,933 distinct terms left after prepossessing the corpus. Some documents were labeled with different category sets in the corpus, so only category set "topics" were used for the experiment. In the experiment, Lewis's training/test split was used. A training set consists of 9,603 documents for training the model, and the test set consists of 3,299 documents for testing the trained model. The same version of dataset in matlab format used in the paper [14][†] was also used in this experiment.

4.2.2 20Newsgroup

The 20 Newsgroup dataset [‡] contains around 20,000 newsgroup documents which are categorized across 20 different newsgroups and every newsgroup relates to some different topics:

"(alt.atheism, comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.guns, talk.politics.mideast, talk.politics.misc, talk.religion.misc)". There are 11,314 training documents and 7,532 testing documents which were used for training the model and testing the trained model.

There are two versions (version 1 and version 2) of this dataset which can be found here [§]. Version 1 consists of 18,774 documents and the feature dimension is 61,188. Version 2 consists of 18,846 documents and the feature dimension is 26,214.

[*] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[†] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html
[‡] http://qwone.com/~jason/20Newsgroups/
[§] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

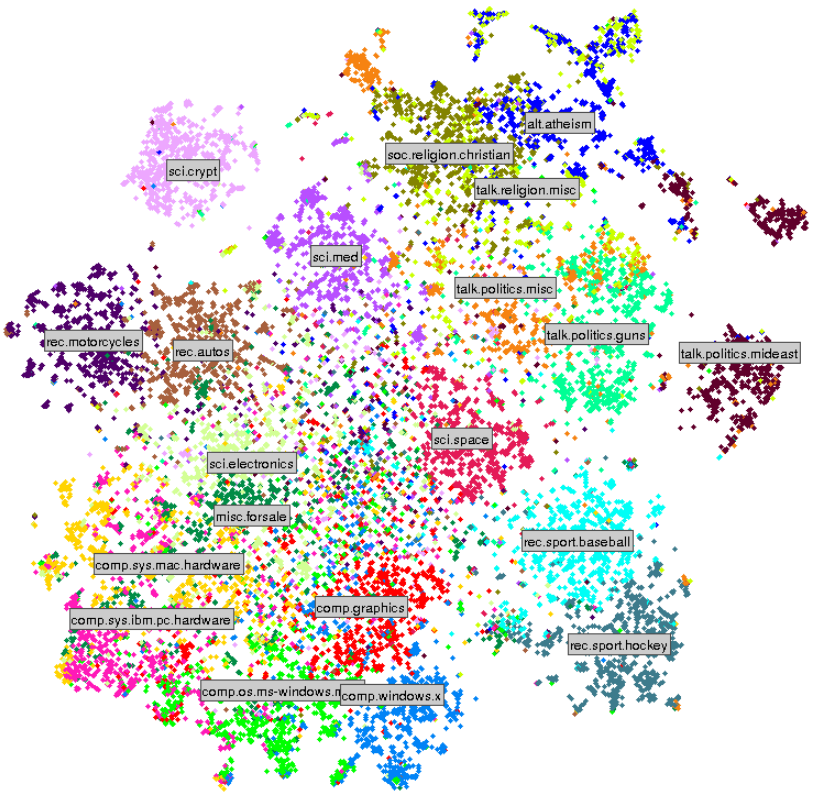


Figure 4.1: 20 Newsgroup Data Visualisation (source) across 20 different newsgroups/topics

4.2.3 MNIST

MNIST handwritten image dataset * is a widely known dataset in the machine learning field. There are 60,000 training samples and 10,000 test samples (grayscale images) with uniform image sizes of 32×32. This image collection is one of the most widely-used datasets for various classification tasks. There are ten classes from 0 to 9. This dataset is a subset of a broader set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

*http://yann.lecun.com/exdb/mnist/

4.2.4 TDT2

TDT2 dataset [*] [†] [‡] (Nist Topic Detection and Tracking corpus) comprises data obtained from 6 sources including two news wires (APW, NYT), two television programs (CNN, ABC), and two radio programs (VOA, PRI). This corpus contains 11,201 documents that are classified into 96 semantic topics or categories. We didn't consider those documents which occurred in two or more categories, so we were left with 9,394 documents and considered the top 30 categories.

4.3 Evaluation Measures

In this thesis we used the following evaluation measures, which are often used to show the performance of ML models.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

where,

- TP is the number of true positives, i.e. the samples that are correctly classified in a certain class;
- FP is the number of false positives, i.e. the samples that are not classified in the class;
- TN is the number of true negatives, i.e. the samples that are not correctly classified in the class;
- FN is the number of false negatives, i.e. the samples that are incorrectly classified in the class.

[*] http://shachi.org/resources/1292
[†] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html
[‡] https://sites.google.com/site/fawadsyed/datasets

4.4 Experimental Setup

The algorithm of QDM for classification is written in the matlab. The CPU server is used for most of the computation for QDM. The hardware setting is listed below in Table 4.1

Table 4.1: Hardware settings for QDM.

Property	modes	CPU	Memory	System	Threads per core
Value	64 bits	40	16G	Windows 10	2

When NN autoencoder is used with QDM on TDT2 dataset, an autoencoder [9, 112] was trained on NVIDIA® Tesla® V100 32Gb; the hardware setting is listed below in Table 4.2. We kept the following parameters in the autoencoder for training: numbers of hidden sizes (10, 25, 50, 100), some training epochs (1000, 2000), the positive saturating linear transfer function for the NN encoder, the L2 weight regularizer, the loss function, and the sparsity regularizer. We trained the autoencoder on different parameters.

Table 4.2: Hardware settings to train Autoencoder (when QDM is trained with NN autoencoder)

Property	modes	GPU	Memory	System	Threads per core
Value	64 bits	Tesla® V100	32Gb	Ubuntu 14.04	2

4.5 Effectiveness Analysis on selected number of Topics

The experiment in this section can also be found in this paper [28, 100] and this section aims to answer RQ1. In this experiment, several measures were used to compute the effectiveness of the proposed quantum detection model including accuracy, precision, recall, and f-measure. SVM and NB are used as baselines in this experiment. Multiple topics are available in the Reuters21578 datasets, which can be referred to as a multi-class classification problem. Such problems are generally solved by decomposing into a binary problem using one-vs-all strategy. One-vs-all strategies are used in this work to deal with multiple topics. In one-vs-all, documents labeled as pertinent to the topic are considered positive samples for each category, while the others are considered negative samples. There are specific criteria for selecting the number of documents and topics in this experiment. There is a subset of 135 topics used in this experiment.

4.5.1 Top 9 Topics with at least 100 documents

This experiment can be found in the paper [28]. For each selected topic, at least 100 documents were pertinent to a topic in the training set and at least one document in the test set. Such selection criteria led to 9 topics for this experiment. The other topics were discarded due to fewer documents for those topics in the training set, which led to the low performance of all classification models, including baselines. Generally, feature dimensions are very high with the documents, which leads to deficient performance if all the features are selected in the training. So it is necessary to choose some top relevant features for the training. In our experiment, χ^2 feature selection [38, 81] was used to select the topmost 100 relevant features or terms for the training. χ^2 is the supervised feature selection approach, which keeps the dependency between the feature set and target variable.

The effectiveness of the classification models can be seen in Table 4.3, 4.4, and 4.5. They show that the quantum detection model outperformed the other models in several cases, i.e., the f-measure for a few topics (topics 7, 8, 9) was higher than the baselines, recall was higher for those same topics (topics 7, 8, 9), precision was higher for those topics (topics 7, 8, 9), and also accuracy was higher for a few topics (topics 5, 6, 8, 9). It should be noted that there are some "NaN" values in Tables 4.3, 4.4, and 4.5. This happened because the algorithms only returned negative documents for some topics, whereas the ground truth consisted mainly of negative documents for the topic.

Table 4.3: Classification performance by Naive Bayes

Topic	Accuracy	Precision	Recall	F-measure
1	0.946	0.994	0.883	0.935
2	0.743	0.840	0.033	0.065
3	0.960	0.833	0.051	0.096
4	0.970	1.000	0.027	0.053
5	0.970	0.000	0.000	NaN
6	0.975	NaN	0.000	NaN
7	0.986	NaN	0.000	NaN
8	0.992	1.000	0.208	0.345
9	0.994	1.000	0.333	0.500

Table 4.4: Classification performance by Support Vector Machine

Topic	Accuracy	Precision	Recall	F-measure
1	0.947	0.970	0.908	0.938
2	0.737	1.000	0.004	0.009
3	0.962	1.000	0.112	0.201
4	0.967	0.200	0.014	0.025
5	0.970	NaN	0.000	NaN
6	0.974	0.000	0.000	NaN
7	0.985	NaN	0.000	NaN
8	0.993	1.000	0.333	0.500
9	0.994	1.000	0.333	0.500

Table 4.5: Classification performance by Quantum Detection Model (QDM)

Topic	Accuracy	Precision	Recall	F-measure
1	0.945	0.987	0.888	0.935
2	0.739	0.910	0.016	0.032
3	0.960	0.750	0.061	0.113
4	0.968	0.250	0.014	0.026
5	0.970	NaN	0.000	NaN
6	0.975	NaN	0.000	NaN
7	0.985	0.500	0.057	0.102
8	0.994	1.000	0.458	0.628
9	0.994	1.000	0.381	0.551

4.5.2 Top Number of selected Topics

The experimental results of this section can be found in the paper [100]. Recall, f-measure, and precision are used to estimate the effectiveness of algorithms. There were several topics selected in this experiment. Generally, feature dimensions are very high with the documents, and it leads to deficient performance if all the features are selected in training. So it is necessary to choose some top relevant features for the training. In our experiment, χ^2 feature selection was used to select the topmost 100 relevant features or terms for the training. χ^2 is a supervised feature selection approach that keeps the dependency between the feature set and the target variable. This test allowed us to select the relevant features corresponding to the given categories. During training, five-fold cross-validation was also used to avoid the overfitting problem.

Figure 4.2 shows that QDM provides high recall for a number of selected topics compared

to baselines, which means that QDM can be used safely where high recall is required in a fixed hyperparameter setting. NB and SVM cannot perform for several topics where QDM provides high recall. In terms of f-measure, it can be seen in Figure 4.3 that QDM outperformed baselines on a number of topics (topics 17, 2, 8, 15, 3, 24, 28, 31, 1). In terms of precision, as can be seen in Figure 4.4, QDM has shown to be effective for several topics as well. The effectiveness of QDM on the range of topics, features, and samples can be found in the later experiments.

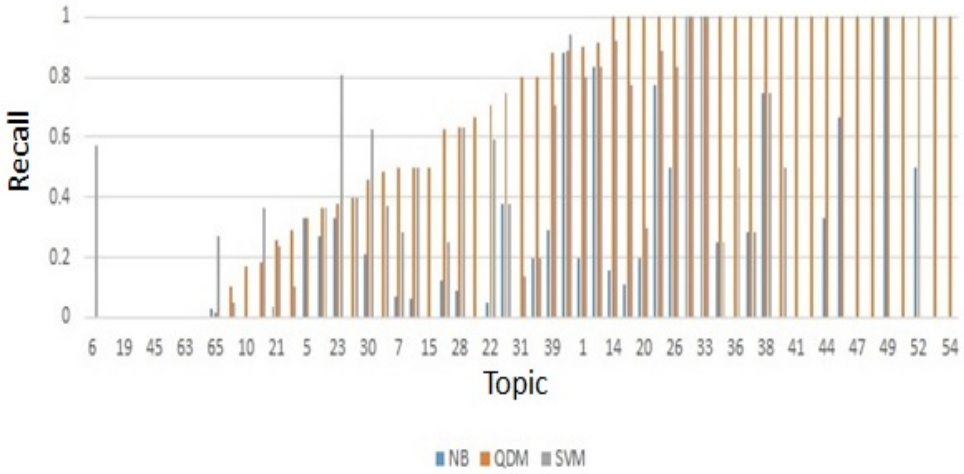


Figure 4.2: Recall Measures of performance for each topic ordered by the measure of the Quantum Detection Model (QDM).

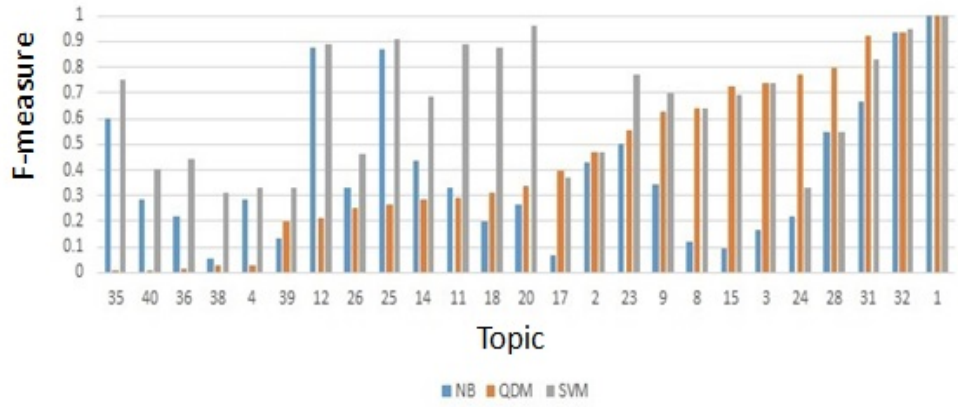


Figure 4.3: F-score Measures of performance for each topic ordered by the measure of the Quantum Detection Model (QDM).

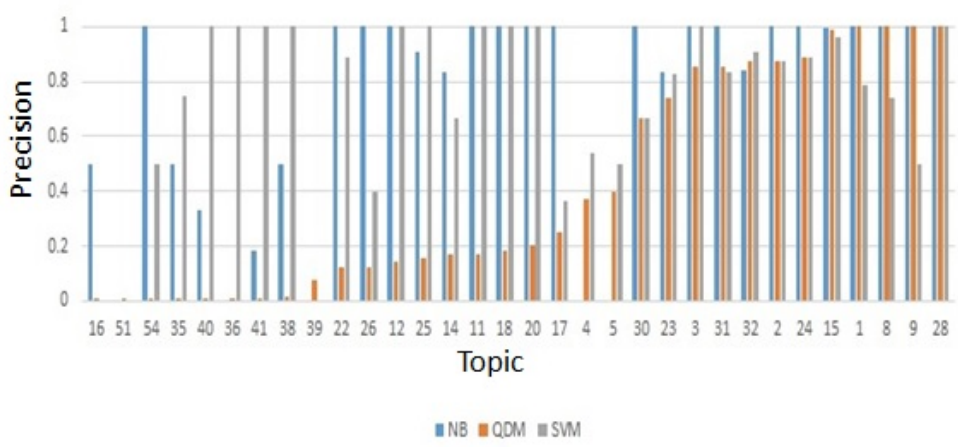


Figure 4.4: Precision Measures of performance for each topic ordered by the measure of the Quantum Detection Model (QDM).

4.6 Effectiveness Analysis on Range of Features

This section aims to answer RQ1 (the diverse range of features and training samples). Some in-depth analysis, such as the comprehensive review of features, is essential to understand the effectiveness of any model. There are many categories, so sometimes the model performs well for one category but performs poorly for another. This issue may arise when a certain range of features is used for the given category during model training. χ^2 feature selection is used to choose a range of features or terms for the model training and testing. The main goal of this test is to select the top relevant features corresponding to the given categories. Therefore, if χ^2 is high for a certain feature, then it is more relevant for that category. χ^2 analysis in Figure 4.8 shows that some terms have higher values and they are more relevant for the given categories. The evaluation measures vary according to the number of selected features, so selecting the top 10 features or top 400 features greatly affects the final performance of the models.

To analyze the changes in the performance, a range of features was selected, specifically the top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200, and 400. These ranges were randomly selected on the basis of the achieved results. In this experiment the baselines, SVM, NB, DT, and KNN, were compared with the proposed QDM. Precision, recall, and f-measure were used to measure the effectiveness of the classification models.

Two datasets were used in this experiment, specifically the 20 Newsgroup and the MNIST handwritten image dataset. All the categories were considered in the analysis of the 20 Newsgroup dataset, but one category was excluded in the MNIST dataset due to its deficient performance in all classification models. There were multiple categories available in the datasets. Such a problem is generally solved by decomposing it into binary problem using one-vs-all strategy. In one-vs-all, samples labeled as pertinent to the categories are considered positive samples for each category, while the remaining samples are considered negative. In this experiment, a 5-fold cross-validation approach was used during the training to avoid the overfitting problem.

4.6.1 Results on the 20Newsgroup Text Corpora

The experimental results of this section can also be found in this paper [102]. QDM outperformed the baselines for several topics or categories as can be seen in Tables 4.6, 4.7, and 4.8. More details about comparisons with baselines are shown in Figures 4.5, 4.6, and 4.7.

The effectiveness of QDM in terms of precision can be seen for several topics, i.e., topics 2, 7, 19, 20, 1, 3, 4, 6, 8, 9, 10, 14, 16 and 18, when the top 5 features were selected (see Figure 4.5). In this case, DT's effectiveness was low and it was unable to perform for 15 out of 20 topics, performing well for only five topics. If the performance of DT is not considered due to its failure to perform on most of the topics, QDM performance in terms of precision was very effective, which can be seen in Figure 4.5. The effectiveness of QDM can also be seen with the top 10 features for a number of topics, i.e., topics 5, 4, 20, 9, 1, 3, 6, 8, 10, 15, 16, and 18, where it outperformed the baselines. The same situation occurred with DT when the top 15 features were selected. DT failed to perform for most of the topics, while QDM was effective for a number of topics, i.e., topics 8, 1, 3, 4, 7, 9, 10, 14, 16, 18, and 20. Similarly, the effectiveness of QDM can be seen for a range of topics, i.e., topics 3, 4, 7, 9, 10, 14, 15, 18, and 20, when the top 20 features were selected. QDM effectiveness deteriorated slightly with an increasing number of features. Feature quality is essential with any classification model, and it is the same with QDM. However, QDM performance was still effective for several topics with an increasing number of features. QDM had a slightly better performance than DT for the top 150 selected features for a range of topics, i.e., topics 17, 5, 2, 3, 7, 9, 10, 11, 14, 16, 18, and 20. The effectiveness of baselines KNN, DT, and SVM was low for most of the topics.

QDM was also effective in terms of recall with an increasing number of features, as can be seen in Figure 4.6. For example, when the top 50 features were selected, then QDM performance proved to be effective for a number of topics, i.e., topics 4, 7, 3, 19, 20, 6, 14, 5, 15, 18, 12, and 11, while DT was still unable to perform. The performance of QDM improved with an increasing number of selected features for most of the topics, as can be seen in Figure 4.6. However, QDM was effective for fewer topics when the top 150 features were selected. KNN was always effective for topic 1.

The effectiveness of QDM in terms of f-measure improved with an increasing number of selected features for most of the topics, as can be seen in Figure 4.7. The performance of QDM started increasing for a range of selected features, i.e., the top 30, 40, 50, 70, and so on. For instance, when the top 50 selected features were considered, QDM outperformed the baselines for most topics, i.e., topics 4, 7, 3, 19, 20, 6, 14, 5, 1, 15, 18 and 11. QDM was effective for more than half of the topics when the top 40, 50, 70, and 100 features were selected. When the top 150 features were selected, QDM outperformed the baselines for six topics. Similarly, QDM performed better for seven to eight topics out of twenty topics

when the top 200 and 400 features were selected. QDM effectiveness and its performance on several topics can be seen in Figure 4.7.

Table 4.6: Precision Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400

5	$QDM_2 = 0.693$ $QDM_6 = 1.00$ $QDM_{18} = 1.00$	$QDM_{19} = 0.818$ $QDM_8 = 1.00$	$QDM_{20} = 0.967$ $QDM_9 = 1.00$	$QDM_1 = 0.972$ $QDM_{10} = 1.00$	$QDM_3 = 1.00$ $QDM_{14} = 1.00$	$QDM_4 = 1.00$ $QDM_{16} = 1.00$
10	$QDM_5 = 0.923$ $QDM_6 = 1.00$	$QDM_4 = 0.923$ $QDM_8 = 1.00$	$QDM_{20} = 0.947$ $QDM_{10} = 1.00$	$QDM_9 = 0.95$ $QDM_{15} = 1.00$	$QDM_1 = 0.971$ $QDM_{16} = 1.00$	$QDM_3 = 1.00$ $QDM_{18} = 1.00$
15	$QDM_8 = 0.918$ $QDM_{10} = 1.00$	$QDM_1 = 0.952$ $QDM_{14} = 1.00$	$QDM_3 = 1.00$ $QDM_{16} = 1.00$	$QDM_4 = 1.00$ $QDM_{18} = 1.00$	$QDM_7 = 1.00$ $QDM_{20} = 1.00$	$QDM_9 = 1.00$
20	$QDM_3 = 1.00$ $QDM_{14} = 1.00$	$QDM_4 = 1.00$ $QDM_{15} = 1.00$	$QDM_7 = 1.00$ $QDM_{18} = 1.00$	$QDM_9 = 1.00$ $QDM_{20} = 1.00$	$QDM_{10} = 1.00$	$QDM_{11} = 1.00$
30	$QDM_{12} = 0.933$ $QDM_{16} = 1.00$	$QDM_3 = 1.00$ $QDM_{18} = 1.00$	$QDM_6 = 1.00$	$QDM_9 = 1.00$	$QDM_{10} = 1.00$	$QDM_{13} = 1.00$
40	$QDM_{12} = 0.933$ $QDM_{16} = 1.00$	$QDM_2 = 1.00$	$QDM_6 = 1.00$	$QDM_8 = 1.00$	$QDM_9 = 1.00$	$QDM_{13} = 1.00$
50	$QDM_4 = 0.745$	$QDM_2 = 1.00$	$QDM_9 = 1.00$	$QDM_{10} = 1.00$	$QDM_{13} = 1.00$	$QDM_{16} = 1.00$
70	$QDM_1 = 0.961$	$QDM_2 = 1.00$	$QDM_3 = 1.00$	$QDM_9 = 1.00$	$QDM_{13} = 1.00$	$QDM_{16} = 1.00$
100	$QDM_1 = 0.961$ $QDM_{13} = 1.00$	$QDM_5 = 0.975$ $QDM_{14} = 1.00$	$QDM_2 = 1.00$ $QDM_{16} = 1.00$	$QDM_3 = 1.00$	$QDM_9 = 1.00$	$QDM_{10} = 1.00$
150	$QDM_{17} = 0.916$ $QDM_{11} = 1.00$	$QDM_2 = 1.00$ $QDM_{14} = 1.00$	$QDM_3 = 1.00$ $QDM_{16} = 1.00$	$QDM_7 = 1.00$ $QDM_{18} = 1.00$	$QDM_9 = 1.00$ $QDM_{20} = 1.00$	$QDM_{10} = 1.00$
200	$QDM_2 = 1.00$	$QDM_{10} = 1.00$	$QDM_{11} = 1.00$	$QDM_{14} = 1.00$	$QDM_{20} = 1.00$	
400	$QDM_2 = 1.00$	$QDM_{10} = 1.00$	$QDM_{14} = 1.00$	$QDM_{17} = 1.00$		

Table 4.7: Recall Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400

5	$QDM_2 = 0.023$	$QDM_{20} = 0.119$	$QDM_{18} = 0.154$			
10	$QDM_4 = 0.061$	$QDM_7 = 0.065$	$QDM_{17} = 0.145$	$QDM_{20} = 0.215$		
15	$QDM_2 = 0.069$					
20	$QDM_8 = 0.111$	$QDM_{19} = 0.177$	$QDM_{16} = 0.560$			
30	$QDM_4 = 0.071$ $QDM_{20} = 0.235$	$QDM_7 = 0.078$ $QDM_{15} = 0.359$	$QDM_2 = 0.120$ $QDM_{12} = 0.463$	$QDM_8 = 0.136$ $QDM_{15} = 0.624$	$QDM_{19} = 0.206$	$QDM_{14} = 0.231$
40	$QDM_4 = 0.089$ $QDM_5 = 0.326$	$QDM_3 = 0.097$ $QDM_{15} = 0.397$	$QDM_7 = 0.102$ $QDM_{18} = 0.486$	$QDM_{19} = 0.235$ $QDM_{12} = 0.531$	$QDM_{14} = 0.254$ $QDM_{10} = 0.586$	$QDM_{20} = 0.254$ $QDM_{11} = 0.651$
50	$QDM_4 = 0.096$ $QDM_{14} = 0.292$	$QDM_7 = 0.104$ $QDM_5 = 0.334$	$QDM_3 = 0.104$ $QDM_{15} = 0.426$	$QDM_{19} = 0.258$ $QDM_{18} = 0.494$	$QDM_{20} = 0.270$ $QDM_{12} = 0.546$	$QDM_6 = 0.276$ $QDM_{11} = 0.654$
70	$QDM_7 = 0.115$ $QDM_{14} = 0.389$	$QDM_4 = 0.127$ $QDM_{18} = 0.513$	$QDM_8 = 0.174$ $QDM_{12} = 0.579$	$QDM_{19} = 0.270$ $QDM_{10} = 0.649$	$QDM_{20} = 0.302$ $QDM_{11} = 0.681$	$QDM_6 = 0.315$
100	$QDM_4 = 0.140$ $QDM_{15} = 0.515$	$QDM_7 = 0.143$ $QDM_{18} = 0.542$	$QDM_8 = 0.278$ $QDM_{12} = 0.602$	$QDM_{19} = 0.287$ $QDM_{11} = 0.714$	$QDM_{20} = 0.334$	$QDM_6 = 0.351$
150	$QDM_4 = 0.150$	$QDM_{13} = 0.254$	$QDM_{19} = 0.30$	$QDM_8 = 0.344$	$QDM_6 = 0.387$	$QDM_{15} = 0.553$
200	$QDM_7 = 0.178$ $QDM_5 = 0.407$	$QDM_3 = 0.179$ $QDM_{15} = 0.589$	$QDM_{13} = 0.287$ $QDM_{18} = 0.603$	$QDM_{19} = 0.309$ $QDM_9 = 0.705$	$QDM_6 = 0.397$ $QDM_{16} = 0.713$	$QDM_8 = 0.397$
400	$QDM_3 = 0.237$ $QDM_{15} = 0.635$	$QDM_7 = 0.264$ $QDM_{18} = 0.659$	$QDM_{13} = 0.343$ $QDM_{16} = 0.768$	$QDM_{20} = 0.406$ $QDM_9 = 0.770$	$QDM_6 = 0.484$ $QDM_{11} = 0.802$	$QDM_8 = 0.513$

Table 4.8: F-measure Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the 20 Newsgroup dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400

5	$QDM_2 = 0.044$	$QDM_1 = 0.202$	$QDM_{20} = 0.212$	$QDM_{18} = 0.267$		
10	$QDM_4 = 0.1141$	$QDM_7 = 0.121$	$QDM_{17} = 0.244$	$QDM_{20} = 0.350$		
15	$QDM_1 = 0.398$					
20	$QDM_8 = 0.198$	$QDM_{19} = 0.291$				
30	$QDM_4 = 0.130$ $QDM_{11} = 0.756$	$QDM_7 = 0.139$	$QDM_{19} = 0.329$	$QDM_{14} = 0.369$	$QDM_{15} = 0.517$	$QDM_{12} = 0.619$
40	$QDM_4 = 0.160$ $QDM_{15} = 0.555$	$QDM_3 = 0.175$ $QDM_{18} = 0.646$	$QDM_7 = 0.176$ $QDM_{10} = 0.731$	$QDM_{19} = 0.362$ $QDM_{11} = 0.776$	$QDM_{14} = 0.399$	$QDM_5 = 0.475$
50	$QDM_4 = 0.171$ $QDM_5 = 0.483$	$QDM_7 = 0.180$ $QDM_1 = 0.534$	$QDM_3 = 0.188$ $QDM_{15} = 0.582$	$QDM_{19} = 0.390$ $QDM_{18} = 0.645$	$QDM_6 = 0.419$ $QDM_{12} = 0.685$	$QDM_{14} = 0.442$ $QDM_{11} = 0.777$
70	$QDM_7 = 0.190$ $QDM_{14} = 0.536$	$QDM_4 = 0.215$ $QDM_{18} = 0.658$	$QDM_8 = 0.281$ $QDM_{12} = 0.707$	$QDM_{19} = 0.398$ $QDM_{10} = 0.774$	$QDM_{20} = 0.443$ $QDM_{11} = 0.794$	$QDM_6 = 0.459$
100	$QDM_7 = 0.226$ $QDM_{15} = 0.649$	$QDM_{16} = 0.302$ $QDM_{18} = 0.678$	$QDM_{19} = 0.399$ $QDM_{11} = 0.813$	$QDM_8 = 0.409$	$QDM_{20} = 0.470$	$QDM_6 = 0.493$
150	$QDM_4 = 0.232$ $QDM_{15} = 0.663$	$QDM_{13} = 0.357$	$QDM_{19} = 0.400$	$QDM_8 = 0.475$	$QDM_6 = 0.527$	$QDM_1 = 0.590$
200	$QDM_3 = 0.275$ $QDM_{15} = 0.665$	$QDM_{13} = 0.386$	$QDM_{19} = 0.399$	$QDM_8 = 0.518$	$QDM_5 = 0.527$	$QDM_6 = 0.533$
400	$QDM_7 = 0.304$ $QDM_{11} = 0.797$	$QDM_3 = 0.324$	$QDM_{13} = 0.401$	$QDM_6 = 0.578$	$QDM_8 = 0.580$	$QDM_{15} = 0.651$

4.6.2 Top Features Visualisation through χ^2 and effect on QDM

The feature visualization in Figure 4.8 gives more insight into the data regarding classification performance. The top features are very important for ML tasks. Few distinct newsgroups were selected because several newsgroups were highly related to each other so it was difficult to obtain useful information. The following topics were selected from 20 Newsgroups Text Corpora by using χ^2: *(a) alt.atheism (Topic 1), comp.graphics (Topic 2), comp.os.ms-windows.misc (Topic 4), comp.sys.ibm.pc.hardware (Topic 5), (b) comp.sys.mac.hardware (Topic 6), comp.windows.x (Topic 7), misc.forsale(Topic 8), rec.autos (Topic 9), rec.motorcycles (Topic 10), (c) rec.sport.baseball (Topic 11), rec.sport.hockey (Topic 12), sci.crypt (Topic 13), sci.electronics (Topic 14), sci.med, sci.space (Topic 15), (d) soc.religion.christian(Topic 16), talk.politics.guns (Topic 17), talk.politics.mideast (Topic 18), talk.politics.misc (Topic 19), talk.religion.misc (Topic 20). comp.os.ms-windows.misc(Topic 3)* is not visualised because of its high χ^2; furthermore, it was and difficult to visualise features for other topics when the Topic 3 was considered.

Figure 4.8 shows that features having a high χ^2 were relevant to the selected newsgroups or topics. For instance, the features *image, jpeg*, and *graphics* were more relevant to *comp.graphics (Topic2)*. The features *god, jesus, atheists*, and *bible* were more relevant to *alt.atheism (Topic 1)* and *talk.religion.misc (Topic 20)*. The features *space, launch, nasa, and*

orbit were more relevant to *sci.space (Topic 15)*. The features *health, medical*, and *patients* were more relevant to *sci.med (Topic 14)* to some extent but their χ^2 values were not very high. As can be seen in Figures 4.7 and 4.6, the features corresponding to *sci.space (Topic 15)* high f-score and recall for most of the ranges of features because those features had very high χ^2 values. Similarly, features with average χ^2 scores, corresponding to *sci.med (Topic 14)* had higher f-measures up to the ranges from top 40-70 features, and had high precision for the top 5, 15, 20, 100, 150, 200, and 400 features, as can be seen in Figure 4.5. In addition, *rec.sport.hockey (Topic 11)* had a high f-measure and recall for almost all ranges of features, which was due to high χ^2 scores for many corresponding features, as can be seen in Figures 4.7 and 4.6. The features corresponding to Topic 11 also had higher frequencies because several other topics were related to Topic 11.

Generally in cases where the features had higher frequencies, the f-measure and recall were also very good, as is the case of Topic 11. The other topics and their corresponding features in Figure 4.8 also provide useful insights into the classification model.

4.6.3 Results on MNIST handwritten image Dataset

The results of this section can also be found in this paper [102]. QDM outperformed the baselines in several categories as can be seen in Tables 4.11, 4.9, and 4.10. QDM proved to be very effective in terms of the recall by outperforming most of the baselines, as shown in Figure 4.9 and Table 4.9. QDM performance varied with the type of datasets used.

QDM was effective in terms of f-measure for all categories except category 0 by outperforming the baselines for a range of features, i.e., the top 5, 10, 15, 20, and 30, as can be seen in Figure 4.10. QDM has also shown effectiveness by outperforming baselines for a range of features up to the top 50 features for categories 1, 2, 5, 6, and 7. If we consider category 6, then QDM always outperformed baselines in all ranges of features up to the top 70 features. Subsequently, QDM performance starts deteriorating with the increasing number of features, i.e., 100, 150, 200, and 400. Thus, QDM performance in terms of f-measure can be very effective up to a certain number of features, and a higher number of features makes it a bit less effective than other classifiers.

QDM effectiveness in terms of precision can be seen in Table 4.11 and Figure 4.11. QDM was very effective in terms of precision for several categories, i.e., categories 0, 1, 5, 6, and 7, when the top 10 features were selected. QDM effectiveness was similar to the other baselines for categories 1, 2, 5, and 6 when the top 15 features were selected. The performance of QDM was identical for categories 5 and 6 when the top 20 features were selected. QDM is

unable to outperform all the baselines with an increasing number of features. However, it outperformed baselines, for instance, when the top 50 were selected for category 1 for SVM. QDM performance was better than SVM for category 3 when the top 70 features were selected. It also outperformed KNN for category 4 when the top 20 features were selected. Therefore, QDM has shown to be more effective for many categories and less effective for some.

Table 4.9: Recall Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400.

5	$QDM_0 = 0.001$ $QDM_6 = 0.001$	$QDM_1 = 0.0007$ $QDM_7 = 0.0066$	$QDM_2 = 0.0014$ $QDM_8 = 0.022$	$QDM_3 = 0.051$	$QDM_4 = 0.049$	$QDM_5 = 0.008$
10	$QDM_0 = 0.001$ $QDM_6 = 0.006$	$QDM_1 = 0.07$ $QDM_7 = 0.038$	$QDM_2 = 0.041$ $QDM_8 = 0.047$	$QDM_3 = 0.147$	$QDM_4 = 0.208$	$QDM_5 = 0.032$
15	$QDM_0 = 0.998$ $QDM_6 = 0.104$	$QDM_1 = 0.240$ $QDM_7 = 0.140$	$QDM_2 = 0.097$ $QDM_8 = 0.237$	$QDM_3 = 0.531$	$QDM_5 = 0.221$	
20	$QDM_0 = 0.00$ $QDM_6 = 0.001$	$QDM_1 = 0.0007$ $QDM_7 = 0.0066$	$QDM_2 = 0.0014$ $QDM_8 = 0.022$	$QDM_3 = 0.051$	$QDM_4 = 0.049$	$QDM_5 = 0.008$
30	$QDM_0 = 1.00$ $QDM_6 = 0.128$	$QDM_1 = 0.283$ $QDM_7 = 0.189$	$QDM_2 = 0.2$ $QDM_8 = 0.721$	$QDM_3 = 0.626$	$QDM_4 = 0.56$	$QDM_5 = 0.522$
40	$QDM_0 = 0.99$ $QDM_8 = 0.911$	$QDM_1 = 0.466$	$QDM_2 = 0.354$	$QDM_3 = 0.792$	$QDM_4 = 0.691$	$QDM_7 = 0.402$
50	$QDM_0 = 0.99$ $QDM_6 = 0.634$	$QDM_1 = 0.554$ $QDM_7 = 0.71$	$QDM_2 = 0.429$ $QDM_8 = 0.943$	$QDM_3 = 0.829$	$QDM_4 = 0.756$	$QDM_5 = 0.761$
70	$QDM_0 = 1.00$ $QDM_6 = 0.801$	$QDM_1 = 0.823$ $QDM_7 = 0.937$	$QDM_2 = 0.828$ $QDM_8 = 0.968$	$QDM_3 = 0.929$	$QDM_4 = 0.878$	$QDM_5 = 0.928$
100	$QDM_0 = 1.00$ $QDM_6 = 0.964$	$QDM_1 = 0.953$ $QDM_7 = 0.999$	$QDM_2 = 0.989$ $QDM_8 = 0.994$	$QDM_3 = 0.95$	$QDM_4 = 0.97$	$QDM_5 = 0.987$
150	$QDM_0 = 1.00$ $QDM_6 = 0.998$	$QDM_1 = 0.989$ $QDM_7 = 1.00$	$QDM_2 = 0.999$ $QDM_8 = 1.00$	$QDM_3 = 0.987$	$QDM_4 = 0.989$	$QDM_5 = 1.00$
200	$QDM_0 = 1.00$ $QDM_6 = 0.999$	$QDM_1 = 0.998$ $QDM_7 = 1.00$	$QDM_2 = 1.00$ $QDM_8 = 1.00$	$QDM_3 = 0.997$	$QDM_4 = 1.00$	$QDM_5 = 1.00$
400	$QDM_0 = 1.00$ $QDM_6 = 1.00$	$QDM_1 = 1.00$ $QDM_7 = 1.00$	$QDM_2 = 1.00$ $QDM_8 = 1.00$	$QDM_3 = 1.00$	$QDM_4 = 1.00$	$QDM_5 = 1.00$

Table 4.10: F-measure Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400.

5	$QDM_1 = 0.0014$ $QDM_7 = 0.012$	$QDM_2 = 0.0028$ $QDM_8 = 0.042$	$QDM_3 = 0.084$	$QDM_4 = 0.092$	$QDM_5 = 0.016$	$QDM_6 = 0.002$
10	$QDM_0 = 0.002$ $QDM_6 = 0.012$	$QDM_1 = 0.129$ $QDM_7 = 0.068$	$QDM_2 = 0.078$ $QDM_8 = 0.088$	$QDM_3 = 0.179$	$QDM_4 = 0.318$	$QDM_5 = 0.061$
15	$QDM_1 = 0.189$ $QDM_7 = 0.094$	$QDM_2 = 0.074$ $QDM_8 = 0.34$	$QDM_3 = 0.268$	$QDM_4 = 0.372$	$QDM_5 = 0.339$	$QDM_6 = 0.05$
20	$QDM_1 = 0.364$ $QDM_7 = 0.18$	$QDM_2 = 0.166$ $QDM_8 = 0.335$	$QDM_3 = 0.299$	$QDM_4 = 0.483$	$QDM_5 = 0.357$	$QDM_6 = 0.187$
30	$QDM_1 = 0.406$ $QDM_7 = 0.197$	$QDM_2 = 0.29$ $QDM_8 = 0.507$	$QDM_3 = 0.279$	$QDM_4 = 0.478$	$QDM_5 = 0.662$	$QDM_6 = 0.224$
40	$QDM_1 = 0.496$	$QDM_2 = 0.368$	$QDM_7 = 0.231$	$QDM_8 = 0.48$		
50	$QDM_1 = 0.502$	$QDM_2 = 0.387$	$QDM_5 = 0.742$	$QDM_6 = 0.663$		
70	$QDM_6 = 0.63$					

Table 4.11: Precision Table where QDM outperforms all the baselines (i.e. QDM_t stands for the performance of QDM for the topic or category t) on the MNIST Handwritten image dataset for all ranges of features, i.e. top 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400.

5	$QDM_6 = 1.00$	
10	$QDM_0 = 0.2$	$QDM_5 = 0.97$
20	$QDM_6 = 0.95$	

Figure 4.5: Precision chart based on top selected features from top 5 to top 400 among KNN, DT, NB, SVM and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents precision.

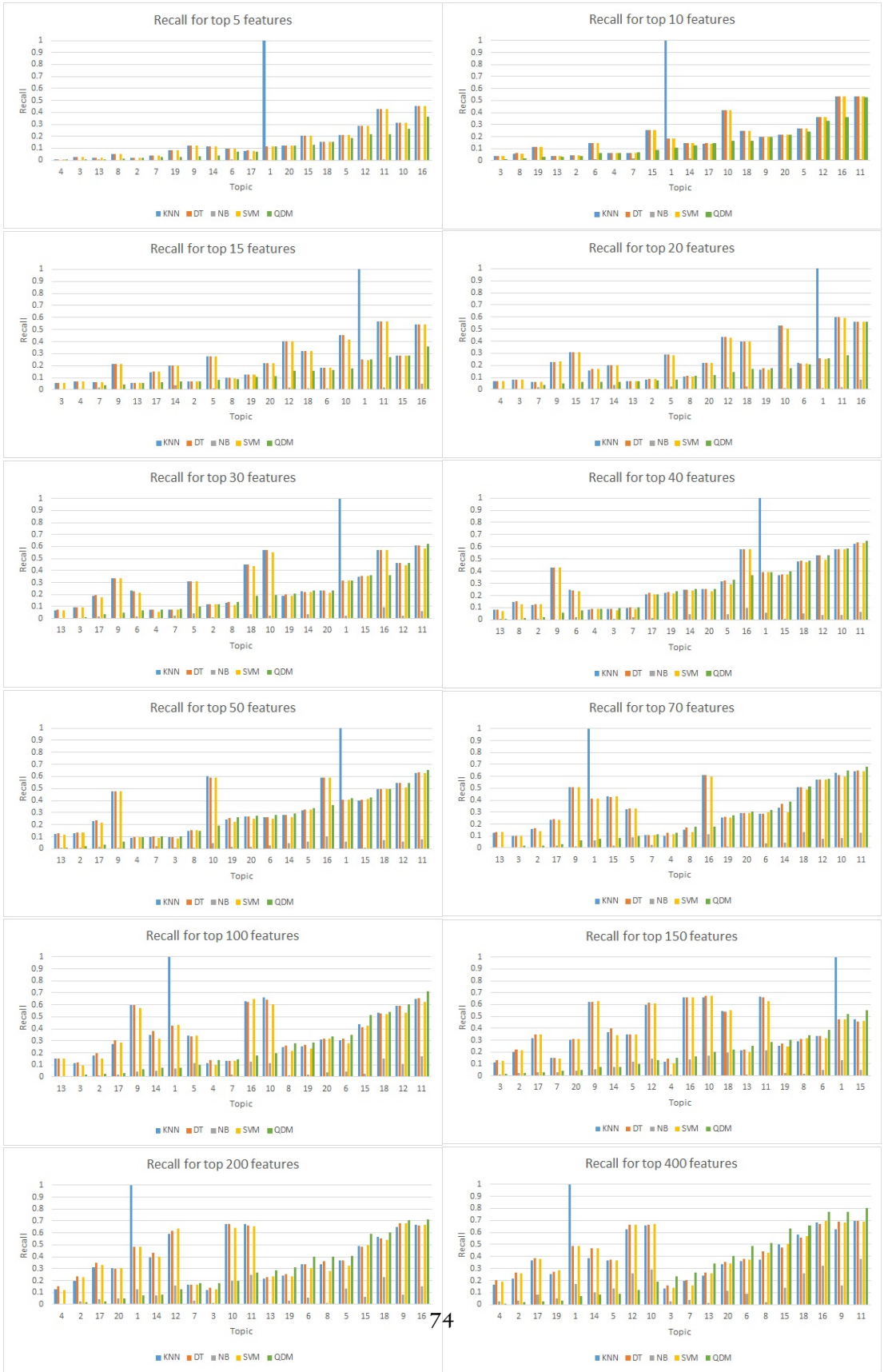


Figure 4.6: Recall chart based on top selected features from top 5 to top 400 among KNN, DT, NB, SVM and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents Recall.

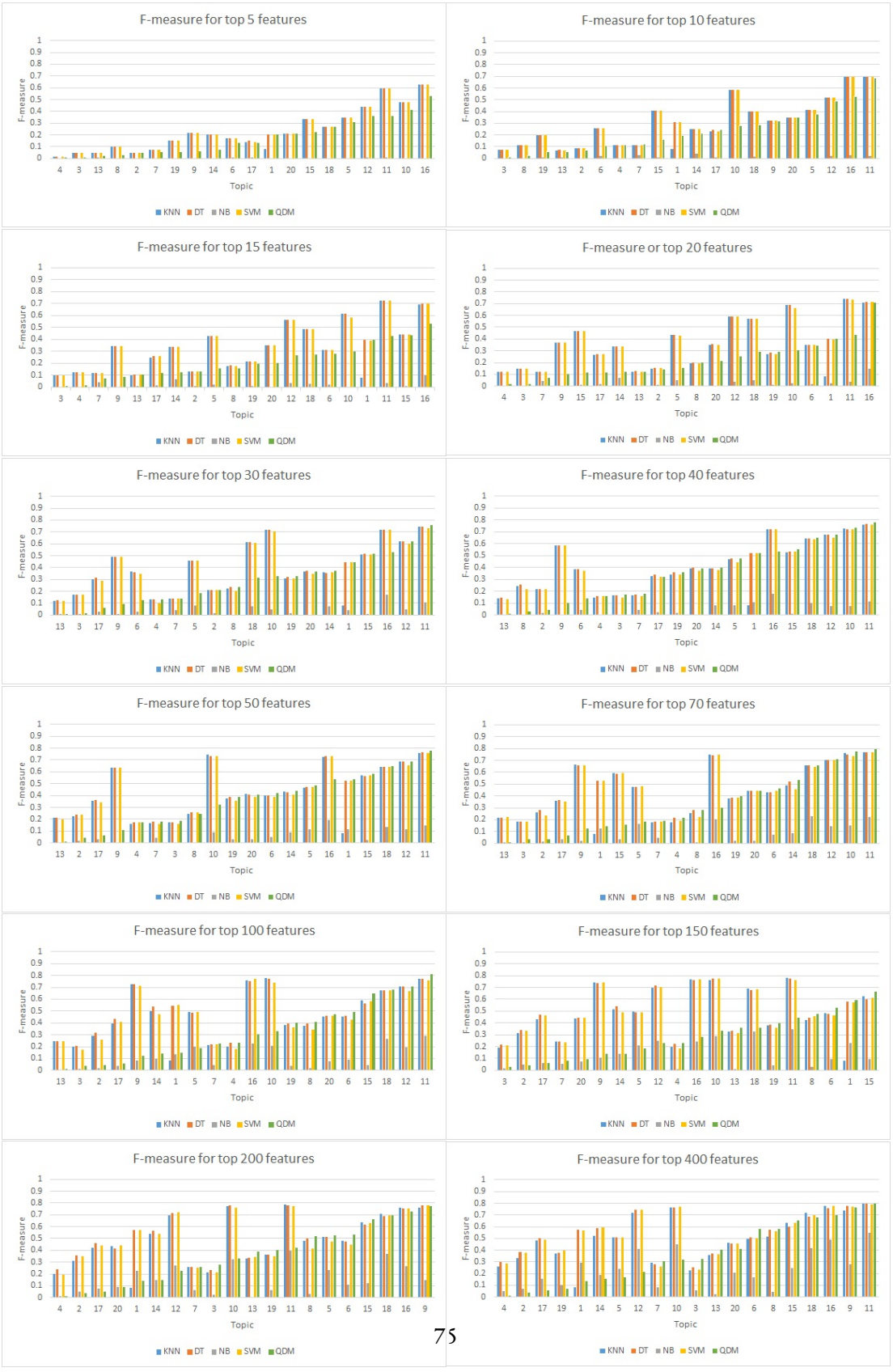


Figure 4.7: F-measure chart based on top selected features from top 5 to top 400 among KNN, DT, NB, SVM and QDM on 20 Newsgroup Text Corpora. X-axis represents Topic and Y-axis represents F-measure.

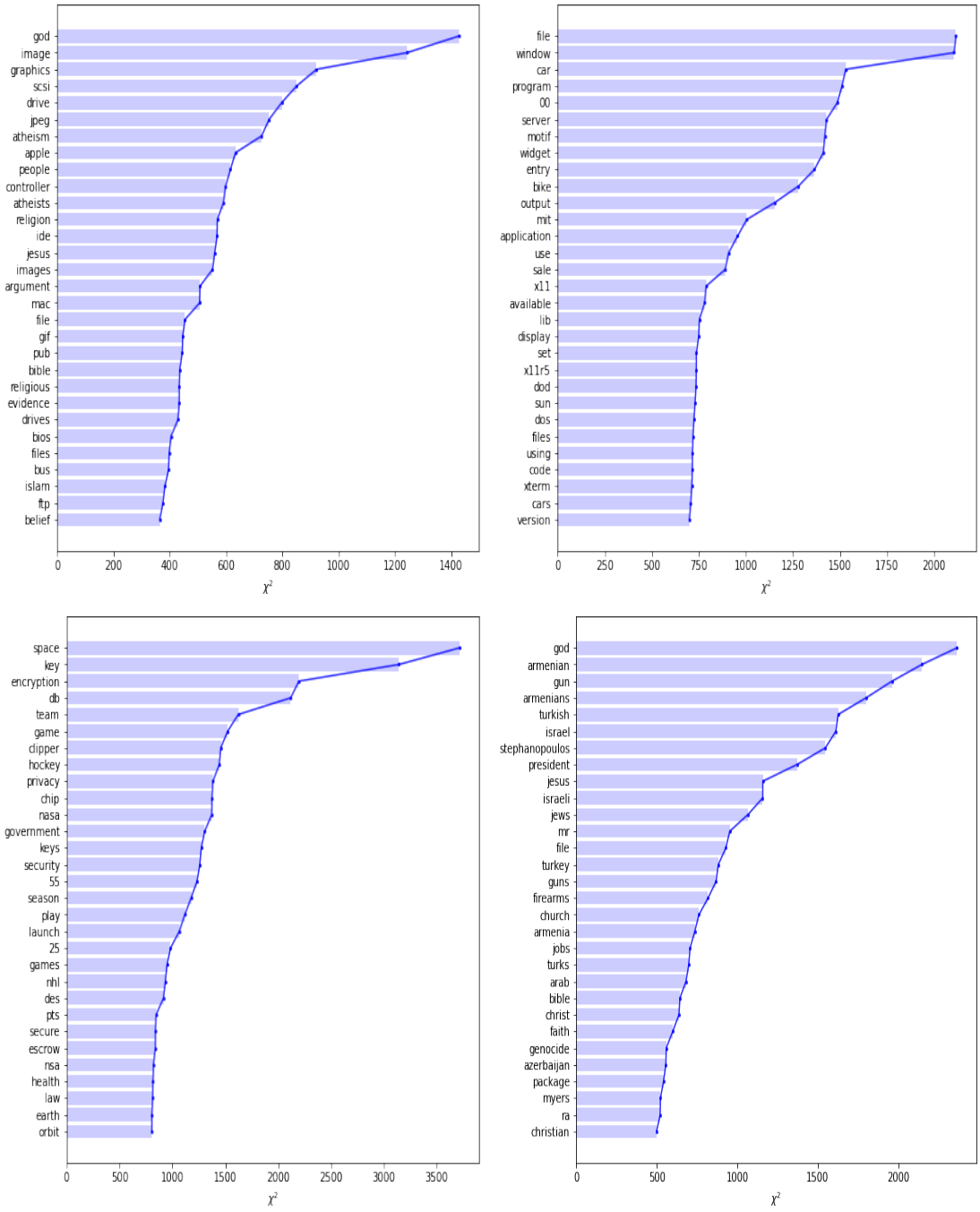


Figure 4.8: Top terms Visualisation corresponding to groups, (a) alt.atheism (Topic 1), comp.graphics (Topic 2), comp.os.ms-windows.misc (Topic 4), comp.sys.ibm.pc.hardware (Topic 5), (b) comp.sys.mac.hardware (Topic 6), comp.windows.x (Topic 7), misc.forsale(Topic 8), rec.autos (Topic 9), rec.motorcycles (Topic 10), (c) rec.sport.baseball (Topic 11), rec.sport.hockey (Topic 12), sci.crypt (Topic 13), sci.electronics (Topic 14), sci.med, sci.space (Topic 15), (d) soc.religion.christian(Topic 16), talk.politics.guns (Topic 17), talk.politics.mideast (Topic 18), talk.politics.misc (Topic 19), talk.religion.misc (Topic 20) on the 20 Newsgroups Text Corpora by using χ^2. comp.os.ms-windows.misc(Topic 3) is not visualised because of its high χ^2; furthermore, it was and difficult to visualise features for other topics when the Topic 3 was considered.

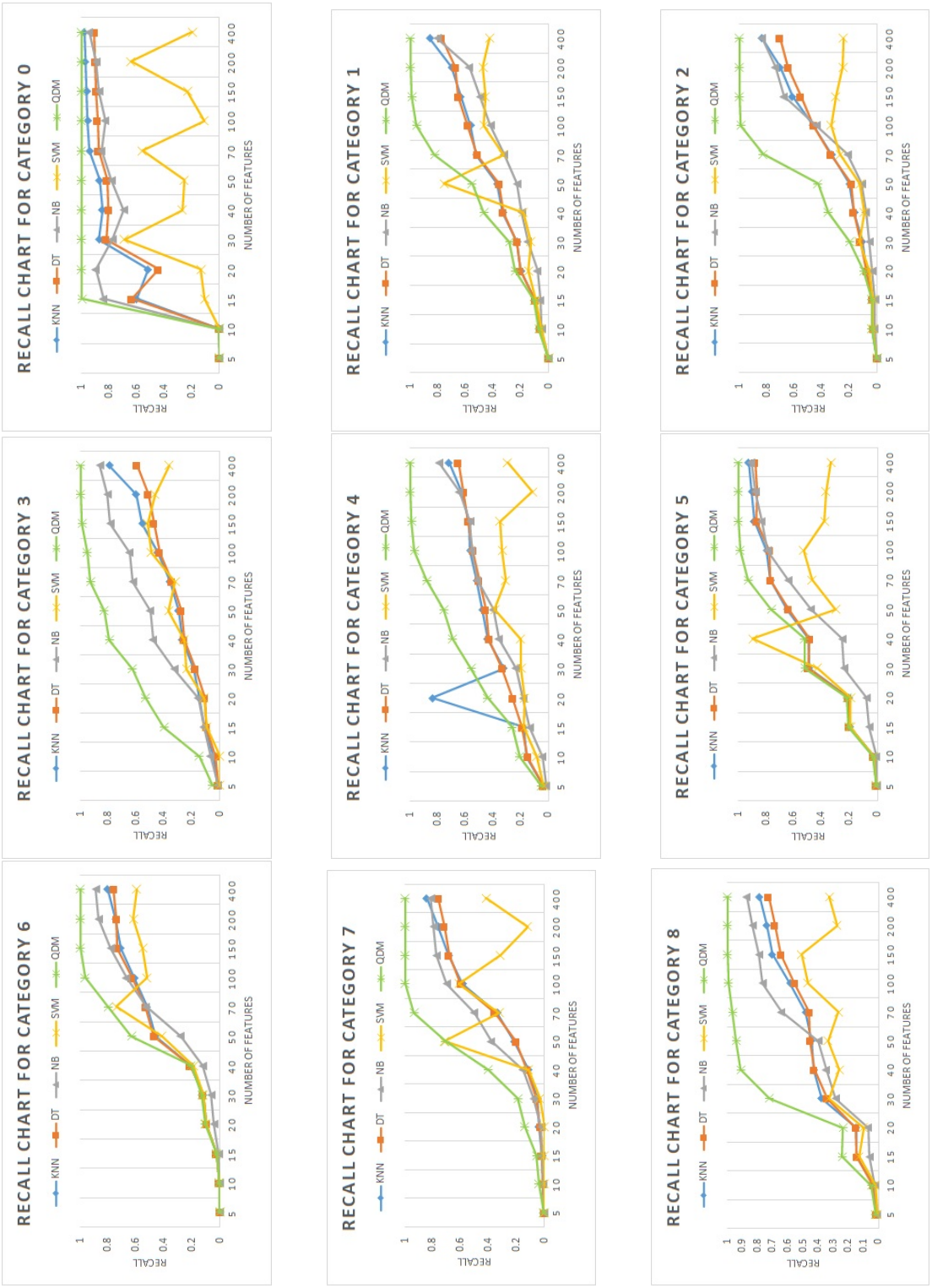


Figure 4.9: Recall chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset.

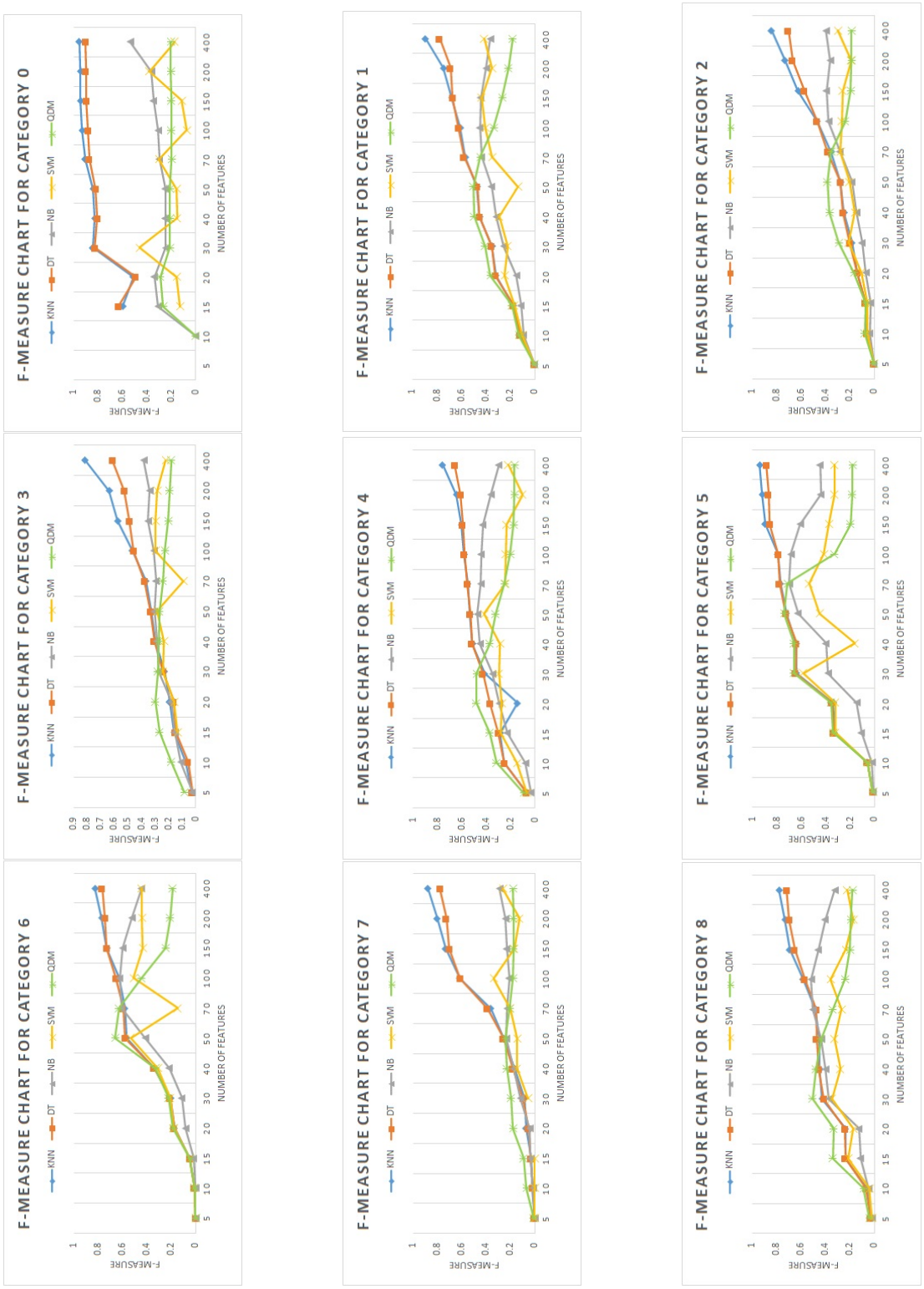


Figure 4.10: F-measure chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset.

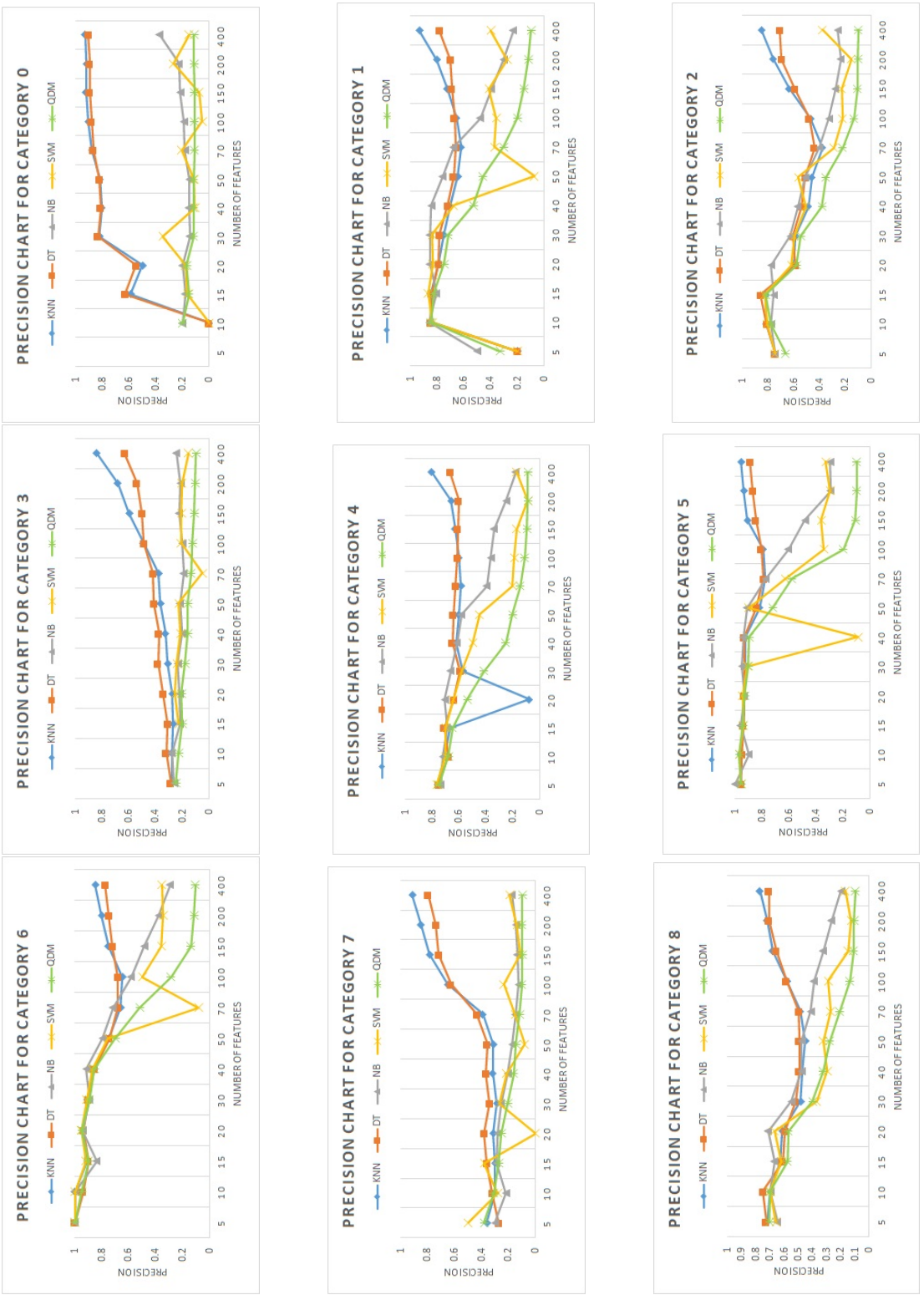


Figure 4.11: Precision chart for each category by changing number of features 5, 10, 15, 20, 30, 40, 50, 70, 100, 150, 200 and 400 among KNN, DT, NB, SVM and QDM on MNIST handwritten image dataset.

The objective of this section is to tackle RQ1 regarding the diverse range of training samples.

The main aim of this study was to understand the performance of QDM and other baselines with fewer training samples on 20Newsgroup (version 2) Text Corpora. The top 100 features and different training samples were selected for analysis. The training samples were randomly selected from the whole sample. The distributions of the selected training samples were 5%, 10%, 20%, 30%, 40%, and 50%, and the rest of the samples were used for the prediction phase. Precision, recall, and f-measure were used as the performance measures. QDM performed better in terms of precision for most of the topics with fewer training samples, and results were similar for recall and f-measure.

QDM outperformed the baselines for topics or categories 1, 2, 8, 9, 11, 12, 15, 17, and 18 in terms of precision with only 5 percent training samples, which can be seen in Figure 4.12. When QDM was trained with 10% training samples with other baselines, it outperformed the baselines for topics 3, 4, 8, 13, 14, and 15, as can be seen in Figure 4.15. The performance of QDM with 20% training samples can be seen in Figure 4.18, where it outperformed baselines for topics 2, 7, 10, 11, 16, 17, 18, 19, and 20. The behaviour of QDM changed with respect to the selected training samples. When QDM performance was checked with 30% training samples as in Figure 4.21, it outperformed the baselines for topics 1, 2, 3, 8, 10, 11, 13, 17, 18, 19, and 20. QDM performance with a 40% training set can be seen in Figure 4.24, where it outperformed the baselines for topics 9, 10, 15, 16, and 20. Furthermore, the performance of QDM with 50% training samples can be seen in Figure 4.27, where it outperformed baselines again in terms of precision for topics, 7, 8, 10, 12, 14, 16, 18, 20.

QDM performance in terms of recall was higher than SVM but lower than other baselines when it was trained with only 5% training samples. When the training samples started increasing for the QDM, then performance also started improving for a number of topics in terms of recall. It can be seen in Figure 4.16 that QDM outperformed the baselines for topics 5, 7, 9, 10, 11, 12, 16, 18, 19, and 20 when 10% training samples were selected. The performance of QDM can also be seen in Figure 4.19 where it outperformed the baselines for topics 3, 4, 6, 8, 12, 13, 14, and 15 when 20% training samples were selected. QDM performance improved with an increasing number of training samples as can be seen in Figure 4.22, and it also outperformed the baselines for topics 4, 5, 6, 12, 14, 15, and 16 when 30% training set were selected. The performance of QDM continued to improve with an increasing number of training samples as can be observed in Figure 4.25, where it outperformed

the baselines for topics 2, 4, 5, 6, 7, 8, 11, 12, 14, 17, 18, and 19 when 40% training sets were used for the training and the rest of them were used for the prediction phase. Furthermore, QDM performance was tested with a 50% training set as can be seen in Figure 4.28, where it outperformed the baselines in terms of recall for topics, 2, 5, 6, 9, 11, 13, 15, 17, and 19.

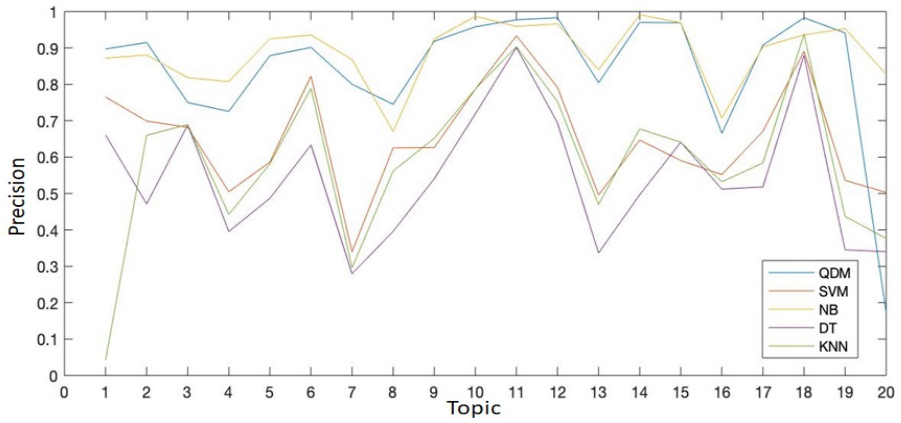


Figure 4.12: Precision of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction.

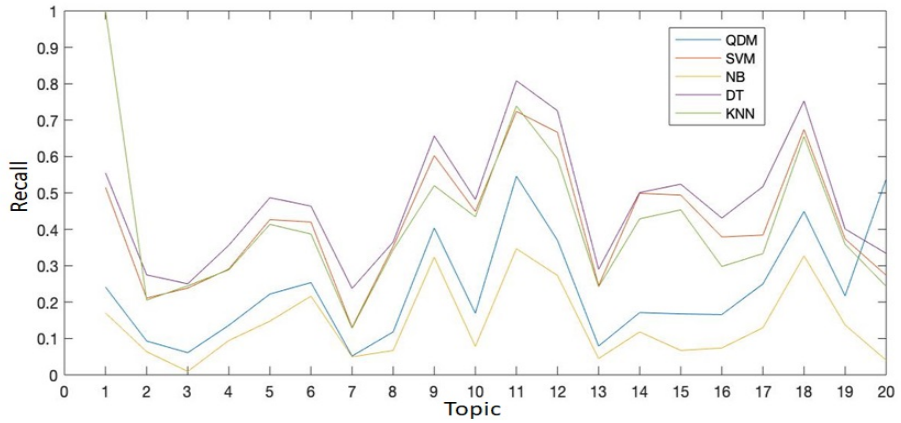


Figure 4.13: Recall of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction.

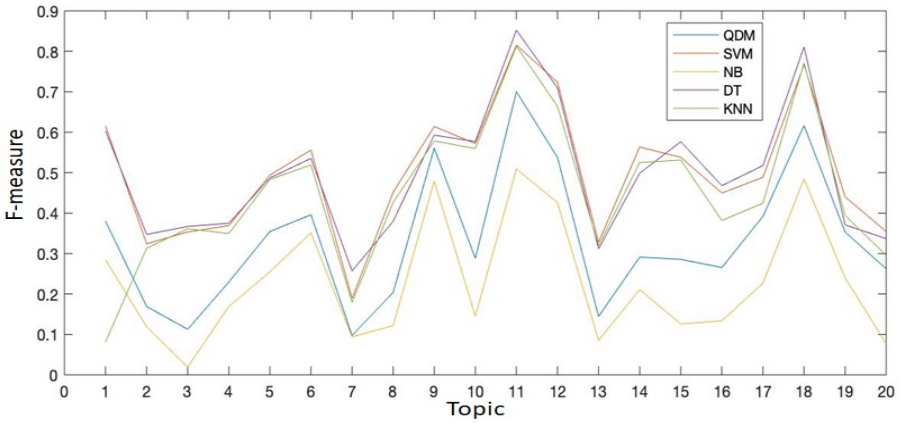


Figure 4.14: F-measure of QDM, SVM, NB, DT, and KNN with 5% training samples and rest for prediction.

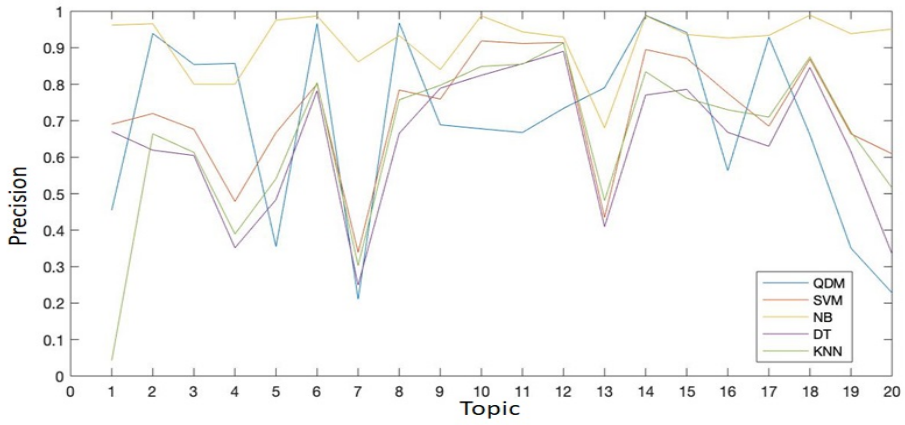


Figure 4.15: Precision of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction.

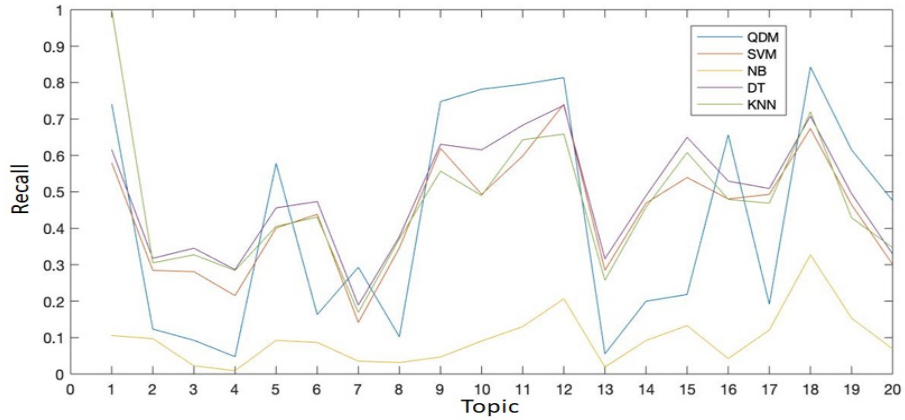


Figure 4.16: Recall of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction.

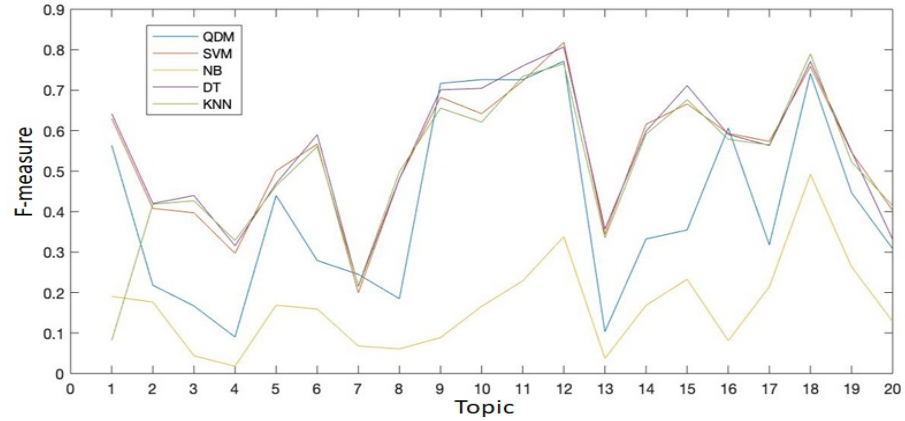


Figure 4.17: F-measure of QDM, SVM, NB, DT, and KNN with 10% training samples and rest for prediction.

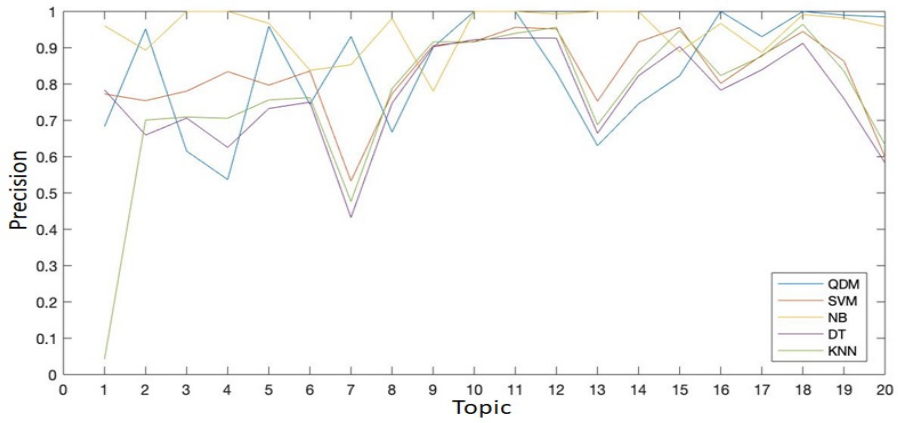


Figure 4.18: Precision of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction.

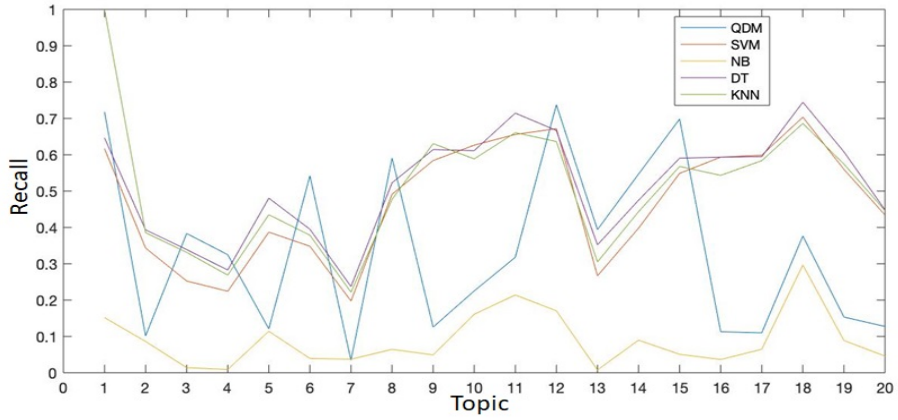


Figure 4.19: Recall of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction.

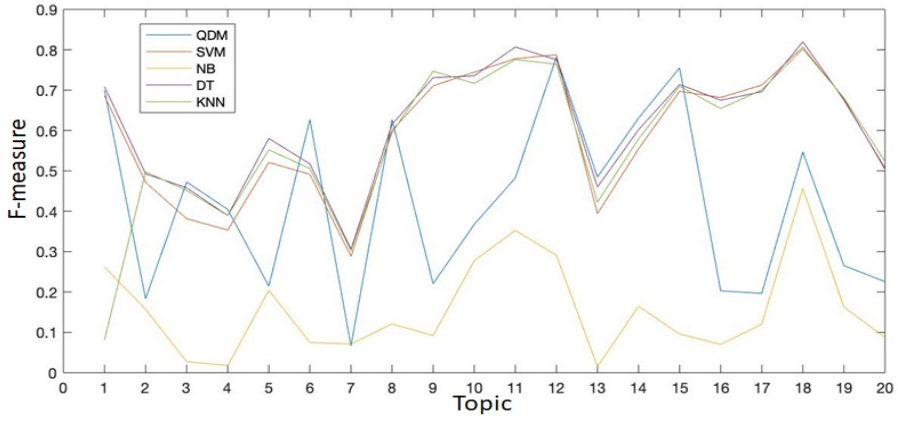


Figure 4.20: F-measure of QDM, SVM, NB, DT, and KNN with 20% training samples and rest for prediction.

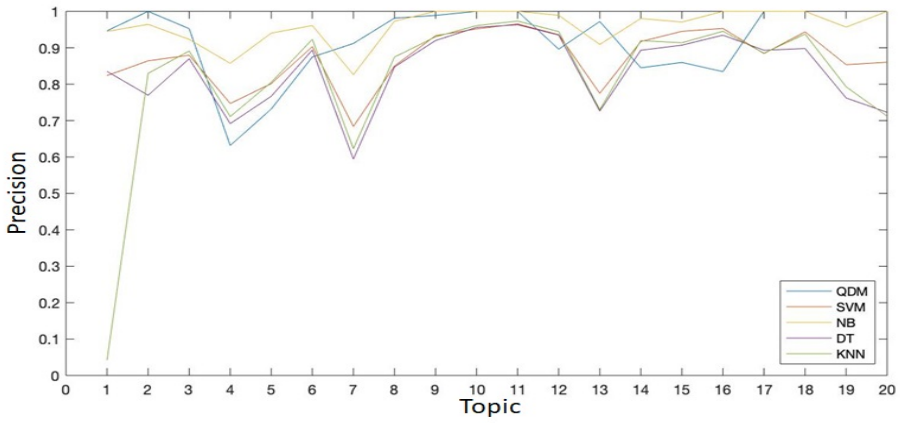


Figure 4.21: Precision of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction.

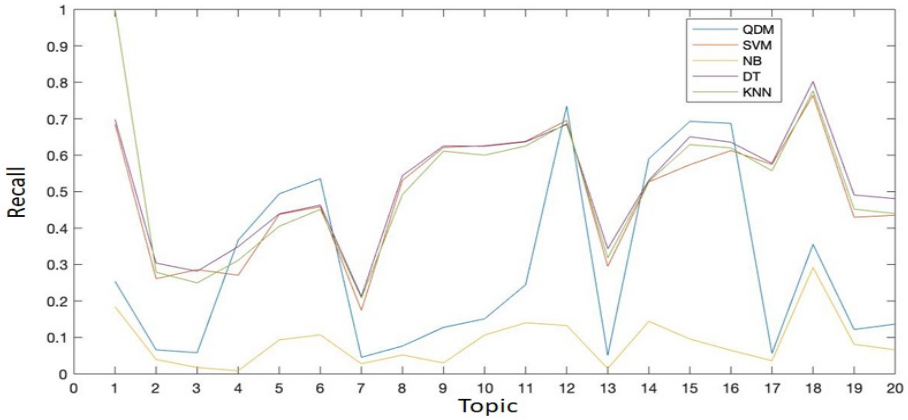


Figure 4.22: Recall of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction.

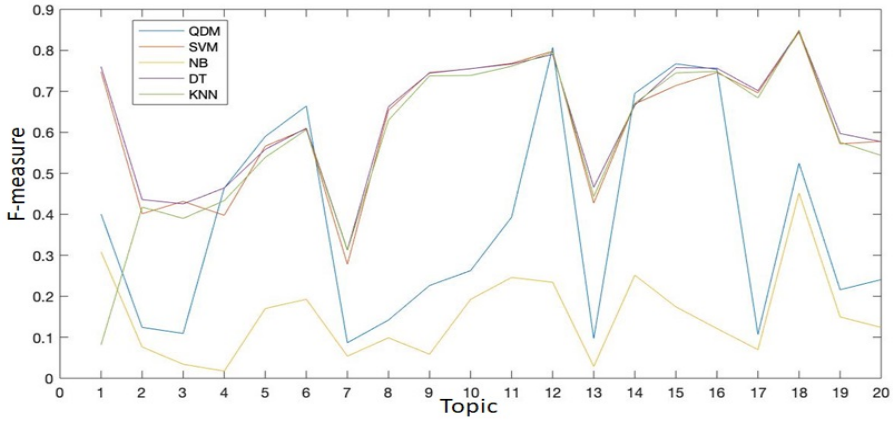


Figure 4.23: F-measure of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction.

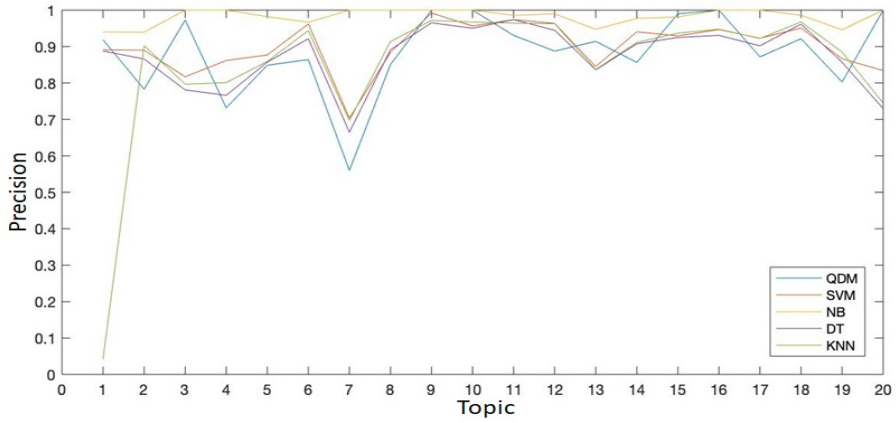


Figure 4.24: Precision of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction.

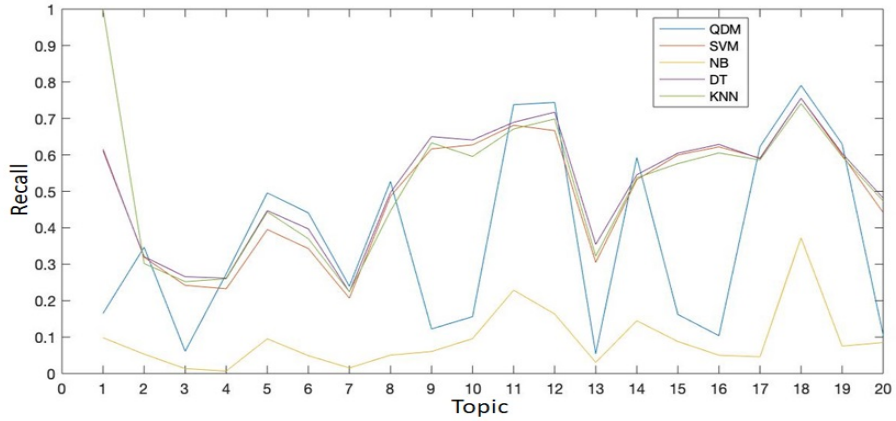


Figure 4.25: Recall of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction.

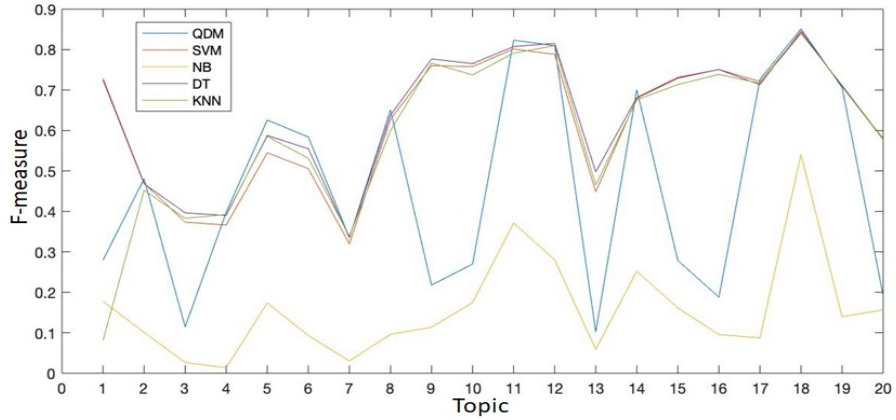


Figure 4.26: F-measure of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction.

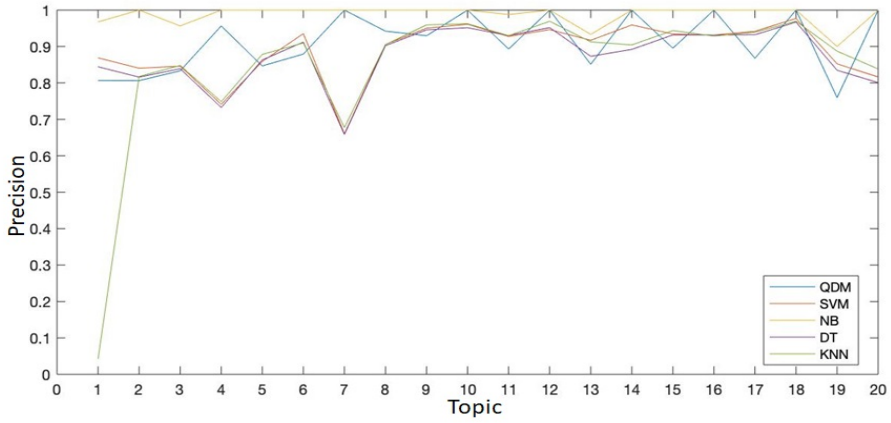


Figure 4.27: Precision of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction.

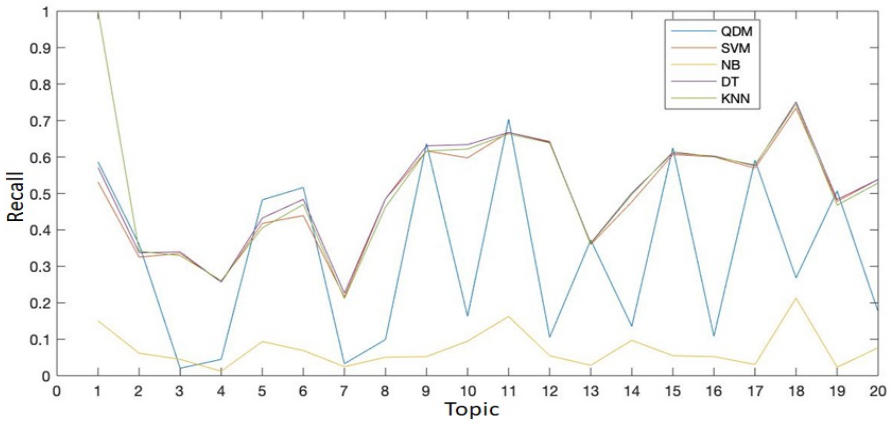


Figure 4.28: Recall of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction.

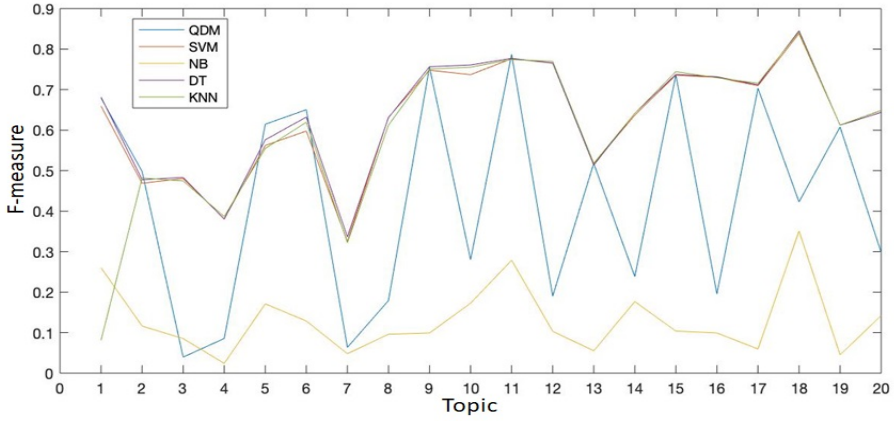


Figure 4.29: F-measure of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction.

4.8 Macro-average and Micro-average analysis for effectiveness

This section focuses on the RQ2 about achieving macro-average and micro-average effectiveness in a one-vs-all strategy scenario.

4.8.1 Description of evaluation parameters

Generally, the effectiveness of classifier is estimated based on the classical notion of precision and recall (see Eq.4.2, 4.3) especially in text classification tasks [92]. Precision (Pr_j) and Recall (R) wrt to category c_j is expressed as follows:

$$Pr_j = \frac{TP_j}{TP_j + FP_j} \tag{4.5}$$

$$R_j = \frac{TP_j}{TP_j + FN_j} \tag{4.6}$$

It is possible to see in Sections 4.5, 4.6, and 4.7 that precision and recall is computed for each topic/category. It is good for the analysis of each category or topic to see the effectiveness at category level. However, the relative average value of all categories can also be useful to determine the global value (Pr and R) for the overall category in order to provide a short explanation. From the logical terminology, Pr can be understood as the *degree of soundness* of the classification model wrt C, and R can be understood as the *degree of completeness* of classification model wrt C. To obtain these global values (Pr and R), two different approaches are used, specifically macro-average and micro-average.

4.8.1.1 Macro-average

In the Macro-average method, local values (precision, recall, f-measure) are computed for each category or topic (see examples in Sections 4.5, 4.6, and 4.7). Then, global values are computed by taking an average of the results of all categories. m is used in Eq. 4.7 and 4.8 to denote macro-average.

$$Pr^m = \frac{\sum_{j=1}^{|C|} Pr_j}{|C|} \tag{4.7}$$

$$R^m = \frac{\sum_{j=1}^{|C|} R_j}{|C|} \tag{4.8}$$

4.8.1.2 Micro-average

In Macro-average method, individual decisions from the different sets are summed to obtain Pr^{μ} and R^{μ}. μ is used in Eq. 4.9 and 4.10 to denote micro-average.

$$Pr^{\mu} = \frac{\sum_{j=1}^{|C|} TP_j}{\sum_{j=1}^{|C|}(TP_j + FP_j)} \tag{4.9}$$

$$R^{\mu} = \frac{\sum_{j=1}^{|C|} TP_j}{\sum_{j=1}^{|C|}(TP_j + FN_j)} \tag{4.10}$$

4.8.2 Analysis on 20Newsgroup Text Corpora

As we can see from Sectiona 4.8.1.1 and 4.8.1.2, the macro-average and micro-average estimate different values, so their results and interpretations are also different. Macro-average values are estimated independently for each category (class or topic), and the average is taken, which can be interpreted as treating all the categories (classes or topics) equally. In contrast, micro-average values are estimated by aggregating the contribution of all categories (classes or topics). When there is a class imbalance problem (too many instances for one category and very few for others, which often occurs when we use one-vs-all strategy to solve the multi-class classification problem), the micro-average is generally preferred. More formally, the macro-average is useful when we want to know the overall classifier performance across different sets of data, and the micro-average is useful when there is variation in dataset size and we need to weight each instance (i.e., documents) equally.

To check the overall effectiveness of QDM and baselines, experimental analysis has been done based on the macro-average and micro-average values. We used 20Newsgroup Text Corpora containing 11,314 training documents and 7,532 testing documents across 20 topics. We kept the same parameters as the top 100 features, using 5 fold cross-validation.

QDM outperformed all the baselines in terms of macro-average precision ($Pr^m = 86.05\%$), which can be seen in Table 4.12. So, the overall precision ($Pr^m = 86.05\%$) across all classes is fairly well. The class imbalance issue is common when using a one-vs-all strategy, so the macro-average value tends to be lower than any classification model. However, QDM can still outperform all the baselines (KNN, DT, SVM, and NB) despite this issue. The confusion matrix of QDM can be found in Figures 4.30, 4.31, 4.32, and 4.33. For a few topics, for example, *topic 7* (hard topic), the precision was lower than 60% for other classification

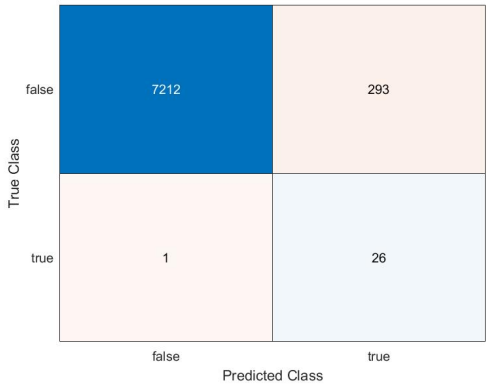
models, including QDM. This low precision for hard topics led to a reduction in Pr^m value when taking the average (treating all classes equally in macro-average), but QDM still outperformed the baselines. *FP* value was higher for topic 7, and this also affected Pr^μ value, which led to some decrements.

The macro-average recall (R^m) value of QDM was lower than KNN, DT, and SVM. There was always a trade-off; when we tuned the model for high precision, we got a low recall. There were some *hard topics* (i.e., topic 1, topic 3, topic 4, topic 12, topic 13, and topic 14,) for QDM to classify in terms of R^m. R^μ value of QDM was also lower than the baselines except for NB.

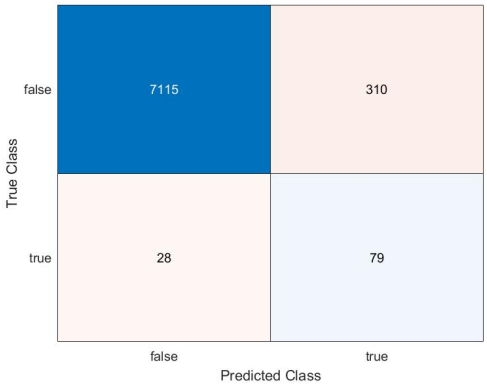
The main reason behind low recall is the large gap between *TP* and *FN* values (see confusion matrix in Figure 4.30, 4.31, 4.32, and 4.33). In these cases, *TP* value of QDM is fairly low and *FN* values are higher, which leads to low R^μ. The reason that R^μ value of KNN is higher because of topic 1, where *TP* value is higher but no *FN*. This *TP* value of KNN allowed to increase the overall R^m value. R^μ value of QDM is higher than NB but lower than others. The micro-average precision (Pr^μ) value of QDM is higher than KNN but lower than others. QDM has shown to be effective in terms of macro-average precision and micro-average precision. It is still possible to improve this performance metric (see Table 4.12) by tuning hyperparameters as discussed in Section 4.9 because obtained results in this Section 4.8 are based on fixed $\lambda = 1$ and a detection boundary of 0.5.

Table 4.12: Macro-average and Micro-average estimates where Pr^m denotes macro-average precision, R^m denotes macro-average recall, Pr^μ denotes micro-average precision, and R^μ denotes micro-average recall.

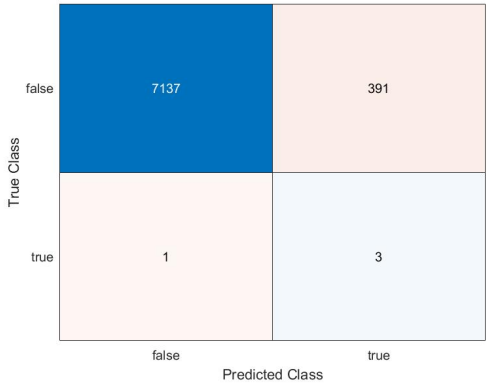
	Pr^m	R^m	Pr^μ	R^μ
KNN	0.8203	0.4484	0.3050	0.4440
DT	0.8370	0.4320	0.8569	0.4318
SVM	0.8585	0.4116	0.8774	0.4114
NB	0.8492	0.0615	0.9688	0.0617
QDM	**0.8605**	0.3139	0.8339	0.3105



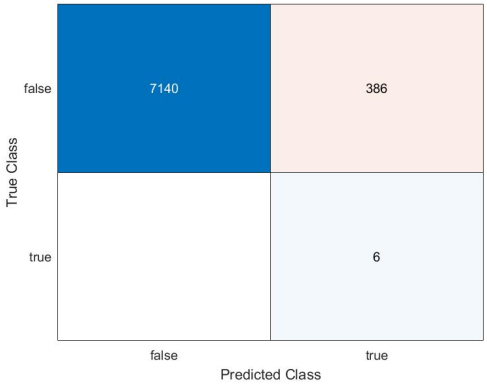
(a) Topic 1



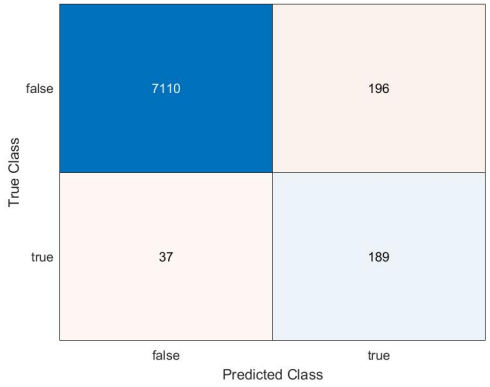
(b) Topic 2



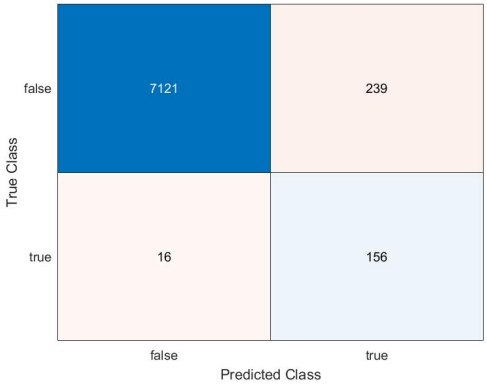
(c) Topic 3



(d) Topic 4

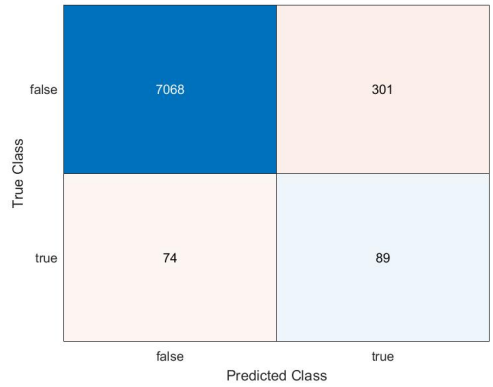


(e) Topic 5

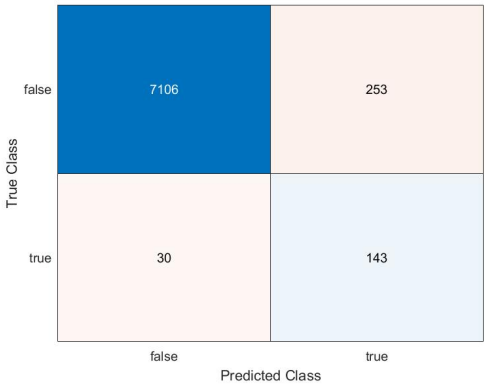


(f) Topic 6

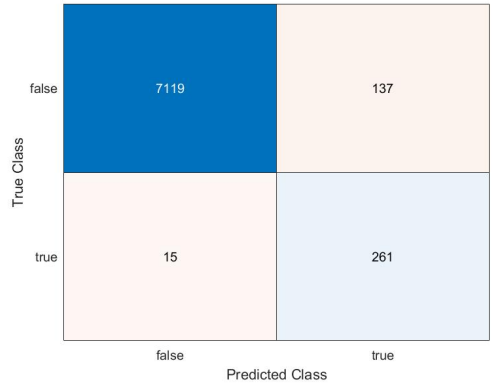
Figure 4.30: Confusion Matrix of QDM for Topic 1 to Topic 6 on 20Newsgroup Text Corpora



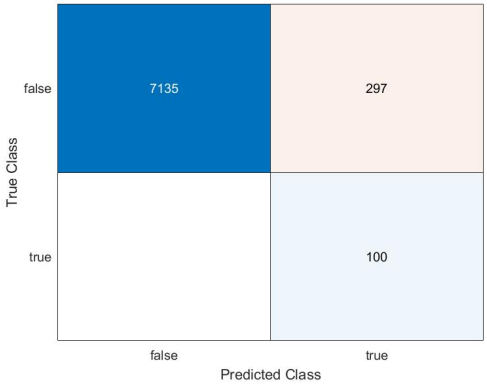
(a) Topic 7



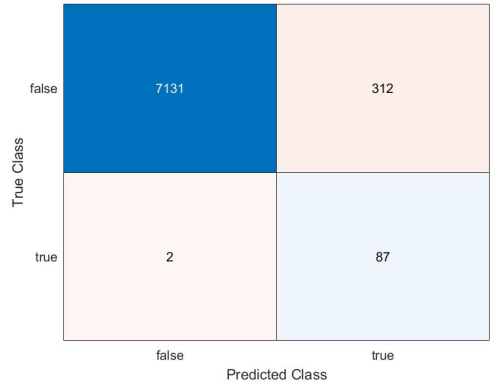
(b) Topic 8



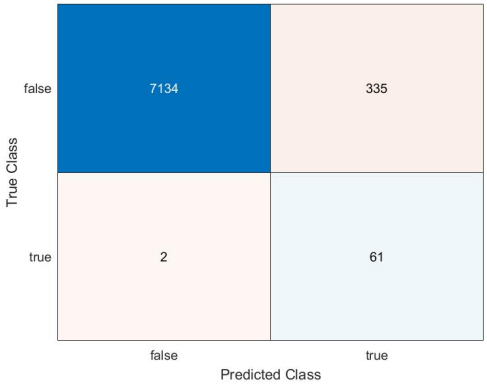
(c) Topic 9



(d) Topic 10

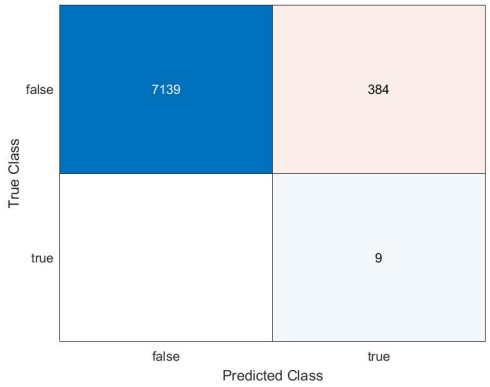


(e) Topic 11

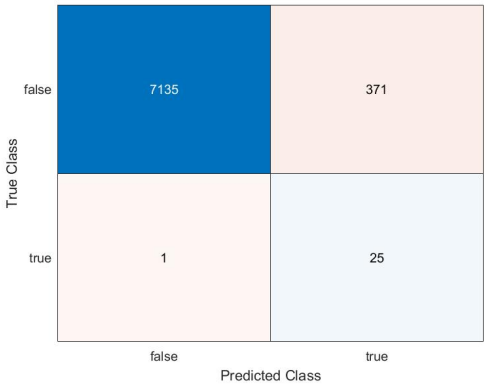


(f) Topic 12

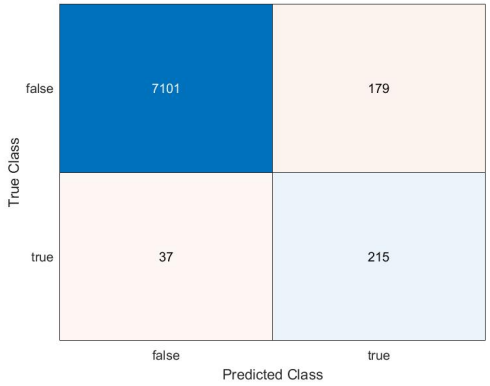
Figure 4.31: Confusion Matrix of QDM for Topic 7 to Topic 12 on 20Newsgroup Text Corpora



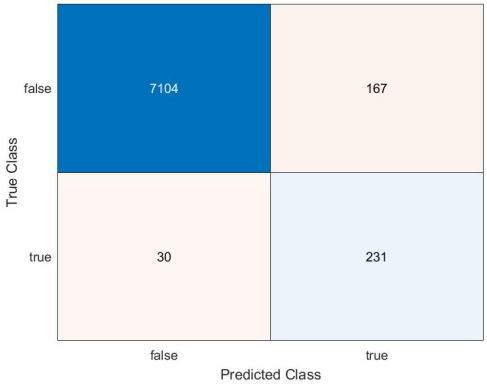
(a) Topic 13



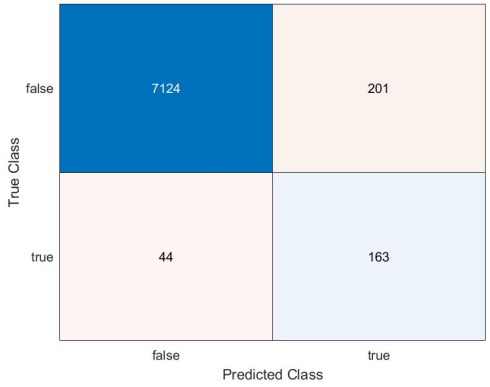
(b) Topic 14



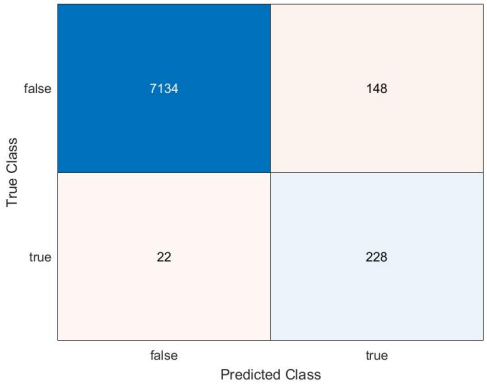
(c) Topic 15



(d) Topic 16

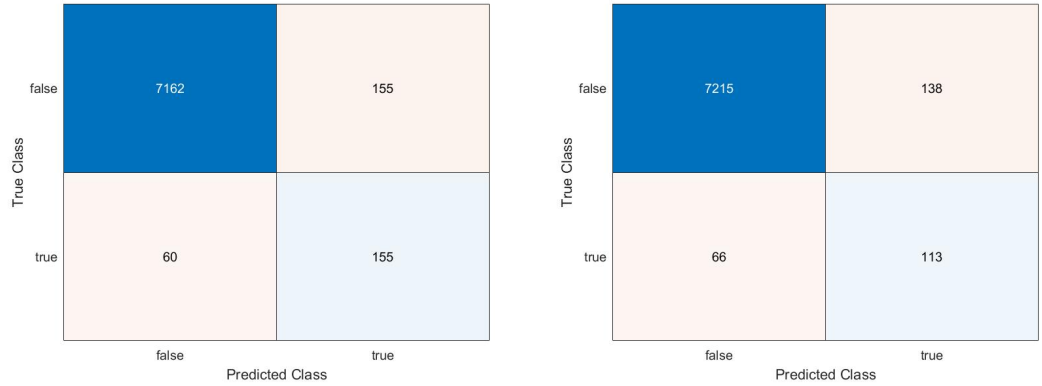


(e) Topic 17



(f) Topic 18

Figure 4.32: Confusion Matrix of QDM for Topic 13 to Topic 18 on 20Newsgroup Text Corpora



(a) Topic 19 **(b)** Topic 20

Figure 4.33: Confusion Matrix of QDM for Topic 19 to Topic 20 on 20Newsgroup Text Corpora

4.9 QDM Performance on different Hyperparameter settings

This study's main aim is to tackle RQ3 which refers to the flexibility of classifier by tuning different hyperparameters. The proposed QDM is quite flexible, in that lambda (λ) and the final detection boundary are the hyperparameters that could be tuned for a particular dataset. QDM has a higher degree of freedom because the parameters can be tuned, leading to improved performance. The parameters are tuned based on the model requirements. Sometimes high precision, recall or f-measure is vital.

4.9.1 For the different value of lambda

This analysis aims to show the QDM performance in terms of precision, recall, and f-measure when the hyperparameter λ is tuned at different values. We used 20Newsgroup Text Corpora for the analysis in this section, and it consists of 11,314 training documents and 7,532 testing documents across 20 topics. QDM and SVM are trained on the same settings by selecting the top 100 features using χ^2 and 5 fold cross-validation. We only used SVM as a baseline for comparison simplicity, but QDM performance is better than other baselines in most cases.

We selected the hyperparameter $\lambda = 0.5$ for QDM, and the obtained results can be seen in Figure 4.34. QDM outperformed SVM for almost all topics or categories except topic 3 and topic 13. QDM precision is very high for almost all topics.

When the hyperparameter λ was tuned to $\lambda = 1.5$, QDM outperformed SVM for all the topics or categories, as shown in Figure 4.35. Figure 4.36 shows that QDM outperformed SVM for all topics except topic 20, in terms of f-measure. As discussed in Section 4.8, *topic 7* is a *hard topic* where precision is lower than 60% for all the classifiers, including SVM, KNN, DT, NB, and QDM. However, the precision of QDM was very high (see Figure 4.34) for this hard *topic 7* when we tuned the value of λ. It is possible to tune the value of λ for those *hard topics*, leading to improvement. Hyperparameter λ is very effective and allows flexibility for QDM to improve the performance.

4.9.2 Effect on QDM performance due to different Detection boundary

The main aim of the analysis in this section is to see the detection boundary effect ($\langle x|\Delta|x\rangle$) when this parameter is tuned. We tuned this parameter on different values, i.e., 0.3, 0.5, and 0.7, to see the behavior of QDM in terms of precision, recall, and f-measure. Figure 4.37 shows the obtained results on a different value of detection boundary. In terms of precision, some topic performances changed after selecting the different values of the detection bound-

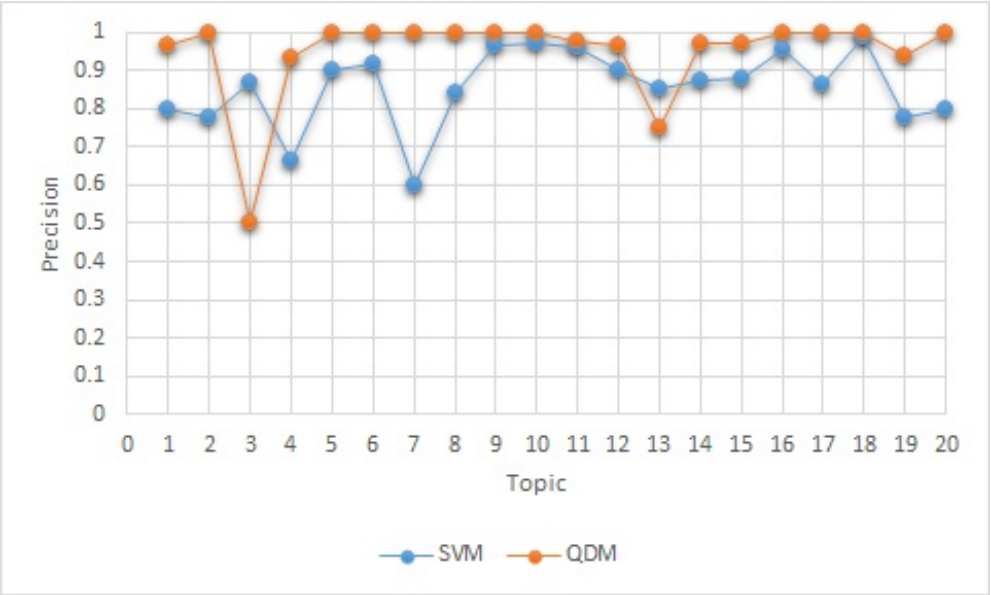


Figure 4.34: Precision for each topic using QDM (when $\lambda = 0.5$) and SVM.

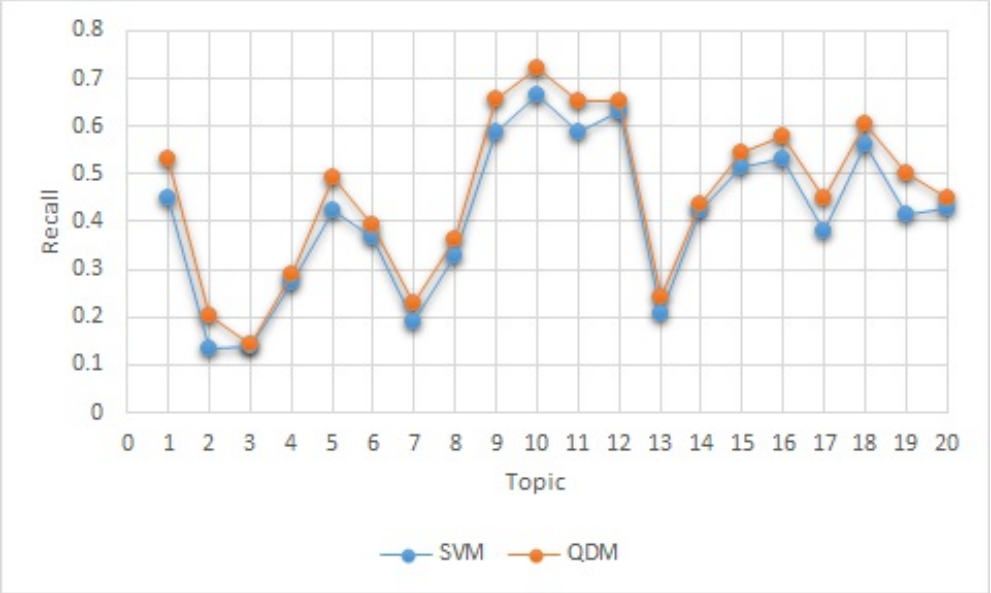


Figure 4.35: Recall for each topic using QDM (when $\lambda = 1.5$) and SVM.

ary. For instance, when $\langle x|\Delta|x\rangle > 0.3$, then topic 1 had higher precision, but this threshold did not suit topic 3 very well. In terms of recall and f-measure, there was some improvement for some topics, i.e., topic 1, 4, 10, 11, 12, 13, 14 when $\langle x|\Delta|x\rangle > 0.3$. These different values of the detection boundary still had an effect on the evaluation metric of QDM to some

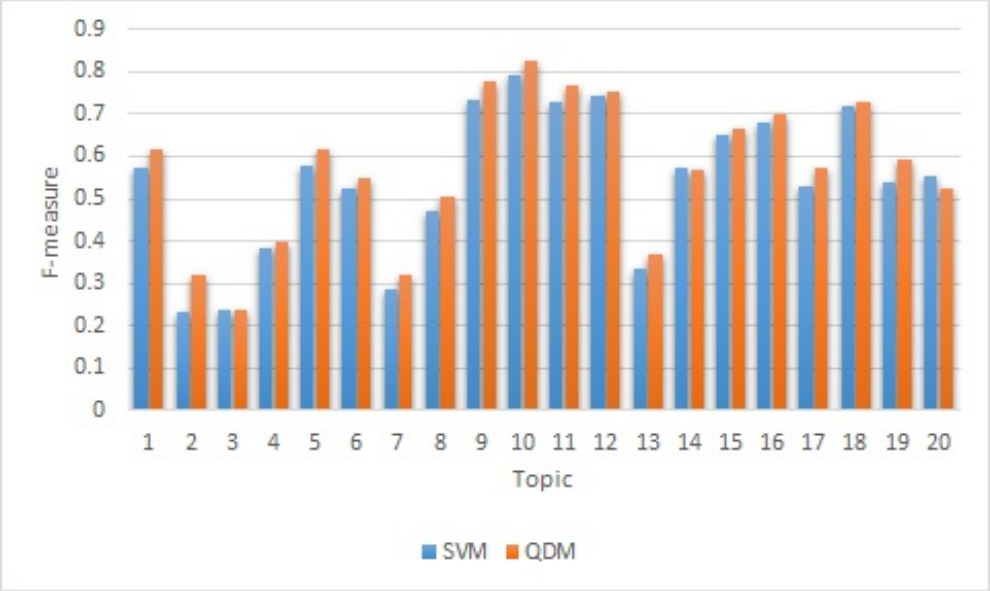


Figure 4.36: F-measure for each topic using QDM (when $\lambda = 1.5$) and SVM.

extent.

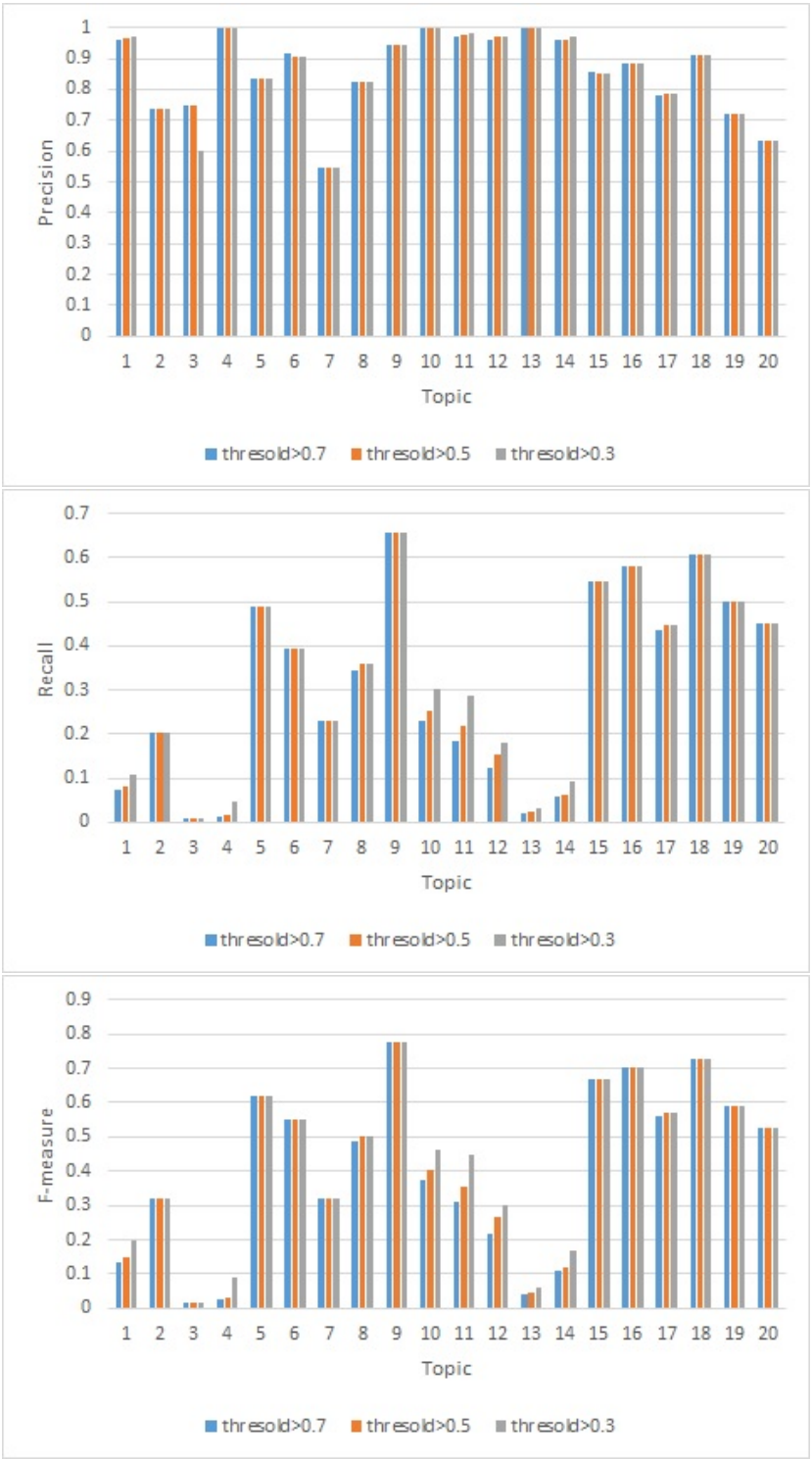


Figure 4.37: Threshold chart of QDM by changing the threshold value $\langle x|\Delta|x\rangle$ > 0.3, 0.5, 0.6, and 0.7 on 20 Newsgroup Dataset. X-axis represents Topics and Y-axis represents Precision, Recall, F-measure in these 3 charts.

4.10 Neural Network Autoencoder as Feature Reduction Method to check QDM effectiveness

A neural network autoencoder is trained for feature transformation and then fed to the QDM model. Later processes are similar to those of supervised ML models.

Autoencoder takes the full feature set of around 37k and transforms it into a low dimension vector. We used the encoded output from the NN encoder as a feature vector and fed it into the quantum detection framework for the training with corresponding labels. Furthermore, it is similar to a supervised learning task for binary classification as it follows one-vs-all.

QDM performance can be seen in Figure 4.38 where the x-axis shows the topics and the y-axis shows the accuracy. In this scenario for example, if we check the performance for $topics$ 1, 18, 27, and 29, then `QDM_25_2K` had 80.9% accuracy which was higher than the others because the hidden size was 25 with 2000 training iterations. It is possible to improve the accuracy for $topic$ 1 by increasing the number of hidden sizes with the number of training iterations. `QDM_100_1K` had the lowest accuracy for $topic$ 1 due to a smaller number of training iterations. QDM performance followed the same trend for $topic$ 2 and $topic$ 1, where the hidden-size and number of training iterations allowed to improve the accuracy. If we check the accuracy of these $topics$ 3, 5, 6, 12, 13, 14, 15, 17, 19, 20, 21, 24, 26, and 29, `QDM_10_1K` had better performance than the others because of the lower hidden size with iterations. Again if we consider $topics$ 4, 6, 9, 11, 12, 22, 23, and 30, `QDM_25_1K` performance was better than the others and had hidden size of 25 and 1000 training iterations. `QDM_10_2K` shows that it is possible to improve the accuracy of these $topics$(7, 8, 10, 13, 16, 20, 25, 28, and 29) by increasing the number of iterations while keeping the hidden sizes low. These different trends for the different topics from Figure 4.38 also show the effect of topicality, which may be a very interesting area for further exploration as well.

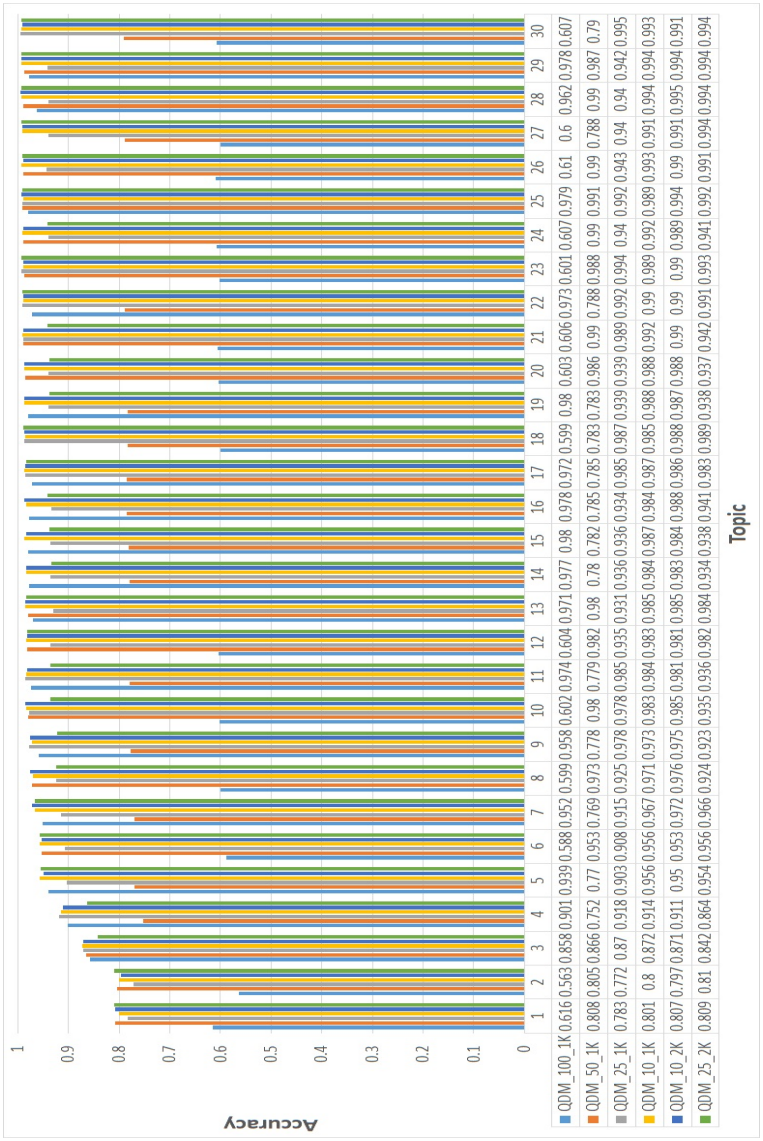


Figure 4.38: QDM Performance on different parameters. `QDM_100_1K` means parameters consist of a hidden-size of 100 with 1000 maximum number of training epochs or iterations. `QDM_50_1K` means parameters consist of a hidden-size of 50 and 1000 epochs. `QDM_25_1K` means parameters consist of a hidden-size of 25 and 1000 epochs. `QDM_10_1K` means parameters consist of a hidden-size of 10 and 1000 epochs. `QDM_10_2K` means parameters consist of a hidden-size of 10 and 2000 epochs. `QDM_25_2K` means parameters consist of a hidden-size of 25 and 2000 epochs.

4.11 Comparison of QDM with Neural Network based classification model

The main aim of this section is to compare QDM with the neural-based state-of-the-art models. 20Newsgroup text corpora are used in this experiment with the same experimental parameters used in previous experiments. Different feature size are considered, and the values of λ is also tuned for QDM. Accuracy is used as an evaluation metric. χ^2 is used to select the top features, and those features are used for the QDM and Artificial Neural Network (ANN).

We used ANN as a neural-based classifier, a two-layer feedforward network consisting of different hidden layer sizes. $Sigmoid$ function was used as an activation function and $softmax$ for the final classification. We set the number of epochs to be 100 for each ANN model. For fair comparison, we fed the same features obtained from χ^2 to the network. QDM outperformed ANN-1 and ANN-2 with the all feature sizes up to 1000, when the value of λ was tuned, which can be seen in Table 4.13. QDM performance was almost the same as ANN-3 when the selected feature size was 2000. Table 4.13 shows that when feature size is around 4000, then QDM performance deteriorates, and ANN-4 has higher accuracy. The ANN model had fairly high accuracy because the features obtained from χ^2 were fed to the ANN network for further processing.

QDM is a non-neural-based model that is based on the statistics of training samples. Like other non-neural-based models, the QDM performance starts deteriorating when the data dimensionality starts increasing. It becomes a bit difficult for classical models to handle large-size data. There were some *hard topics* to classify for QDM, such as topics 5, 6, 7, 9, 11, 12, 15 and 16 when $\lambda = 1$ for selected feature size of 2000. However, performance of these *hard topics* was improved to some extent by the proper tuning of λ. Neural-based models work better when the data size is large, but that also requires high-performance GPUs.

Several different SOTA models were also used for the comparison [50, 8, 109, 123, 15]. There is a difference in the results due to the different parameter settings like feature engineering, hidden layer sizes, number of epochs, etc. SCNN & TSCNN [8] obtained 91.729 % accuracy on the feature size of 10,000. Deep Belief Network (DBN)+Softmax(2) [50] achieved 86% accuracy on the feature size of 2000. SATL [109] is a transfer learning-based approach in cross-domain corpora that performs text classification. SATL [109] achieved an accuracy of 95.62% for the six categories. Camacho et al. [15] obtained an accuracy of 89.08% with the Convolutional Neural Network (CNN) and 90.09% with the CNN+Long

Short-Term Memory (LSTM) for six categories. Overall, QDM works fairly well for some limited feature sizes, but performance starts deteriorating and computation also takes a long time when the feature size is large. The large feature size is one limitation of QDM, as can be seen in Table 4.13.

Table 4.13: Performance Comparison of QDM with neural based models for topic categorization on 20Newsgroup Text Corpora.

Model	**Accuracy**
QDM-1 (feature size=500, $\lambda = 1$)	93.72
QDM-2 (feature size=500, $\lambda = 0.5$)	96.12
QDM-3 (feature size=1000, $\lambda = 1$)	87.13
QDM-4 (feature size=1000, $\lambda = 0.5$)	96.50
QDM-5 (feature size=2000, $\lambda = 1$)	67.88
QDM-6 (feature size=2000, $\lambda = 0.5$)	95.44
QDM-7 (feature size=4000, $\lambda = 0.5$)	84.76
ANN-1 (epochs=100, hidden layer size=500)	95.73
ANN-2 (epochs=100, hidden layer size=1000)	96.03
ANN-3 (epochs=100, hidden layer size=2000)	95.71
ANN-4 (epochs=100, hidden layer size=4000)	95.65
DBN + Softmax(1) [50] (feature size = 2000)	68.71
DBN + Softmax(2) [50](feature size = 2000)	85.57
SCNN [8] (feature size = 10000)	82.76
TSCNN [8] (feature size = 10000)	91.729
SATL [109] (for six categories)	95.62
CNN [15] (for six categories)	89.08
CNN+LSTM [15] (for six categories)	90.09

4.12 Efficiency Analysis

This section focuses on efficiency analysis to tackle RQ4 in order to see the computational cost of QDM compared to other baselines.

4.12.1 Computational time on the 20Newsgroup Text Corpora on range of features

The computation time of the baselines and QDM on 20Newsgroup (version 1) Text Corpora can be found in Table 4.14. QDM performance was also quite good in terms of computational speed. QDM took less time in terms of computation than baselines like KNN and SVM, based on the number of selected features. DT and NB were only slightly faster than QDM. Computation took longer time when the number of features was higher, which was the case with QDM and the others.

Table 4.14: Computation Time in seconds for KNN, DT, NB, SVM, and QDM on the 20 Newsgroup Text Corpora

Features	KNN	DT	NB	SVM	QDM
5	5.2256	0.2455	0.2304	10.466	**0.8577**
10	8.0862	0.2795	0.2823	10.1312	**0.8983**
15	7.3662	0.3689	0.3456	10.4679	**0.9458**
20	8.6009	0.4472	0.4567	10.5934	**1.136**
30	12.3551	0.6619	0.5872	10.8999	**1.397**
40	15.6101	0.9297	0.7308	10.9707	**1.7699**
50	20.1709	1.3405	0.8987	11.778	**2.1004**
70	26.823	2.0612	1.1737	12.4184	**2.6618**
100	39.6668	3.4751	1.7378	14.6174	**3.6232**
150	62.1047	6.2484	2.3273	17.0815	**6.5877**
200	86.231	9.7856	3.0617	20.07	**13.237**
400	172.9039	25.7639	6.0116	30.1179	**101.2075**

4.12.2 Computational time on the MNIST Dataset on range of features

The computation time of the baselines and QDM can be found in Table 4.15. In terms of computational speed, QDM performance was much better than KNN and SVM. Generally, computational speed became expensive and it was also a challenge when it came to image datasets. DT and NB performed only slightly better than QDM. When the number

of features was higher, then SVM and KNN took longer than QDM to compute. This efficiency analysis is encouraging and shows that QDM is also the better replacement for SVM and KNN when it comes to efficiency.

Table 4.15: Computation Time in seconds for KNN, DT, NB, SVM, and QDM on the MNIST Handwritten Image Dataset

Features	KNN	DT	NB	SVM	QDM
5	31.7634	0.4756	1.0224	1841.3	**2.0952**
10	48.5518	0.6199	1.0682	2702.7	**2.1392**
15	45.8418	0.7878	1.1197	3659.4	**2.2417**
20	57.7713	1.1635	1.2810	5348.7	**2.6347**
30	81.7262	1.8554	1.4868	5924.3	**3.2154**
40	121.1638	2.8600	1.8492	6984.8	**3.1704**
50	147.7935	3.6617	2.0850	8107.1	**3.7973**
70	215.7735	6.0609	2.5716	11022	**5.6506**
100	321.4240	9.5287	3.3811	17017	**10.0570**
150	486.7824	13.7031	4.7636	25687	**19.2916**
200	639.0312	17.3438	6.0727	32468	**29.1939**
400	1199.4	29.1856	11.1044	54941	**166.8406**

4.12.3 Computational time on the 20Newsgroup Text Corpora for each topic

4.12.3.1 With Top 100 Features

Figure 4.39 shows the computational time for each category or topic for each classifier KNN, DT, NB, SVM, and QDM, on 20Newsgroup (version 2) Text Corpora. The top 100 features were considered in this setting for efficiency analysis based on each topic. KNN and SVM took longer in terms of computation cost as compared to the proposed QDM. For many topics, QDM took less time to compute than all the baselines, i.e., for topics 2, 3, 4, 7, 8, 9, 14, 15, 16, and 19. QDM can also be a replacement for SVM and KNN in terms of efficiency because of its lower computational cost. It also has a lower computational cost than NB and DT for most of the topics.

4.12.3.2 With Different training samples

The computational time of QDM and the baselines can be seen in Figures 4.40, 4.41, and 4.42, which show that KNN always takes a longer time to compute. The efficiency of QDM

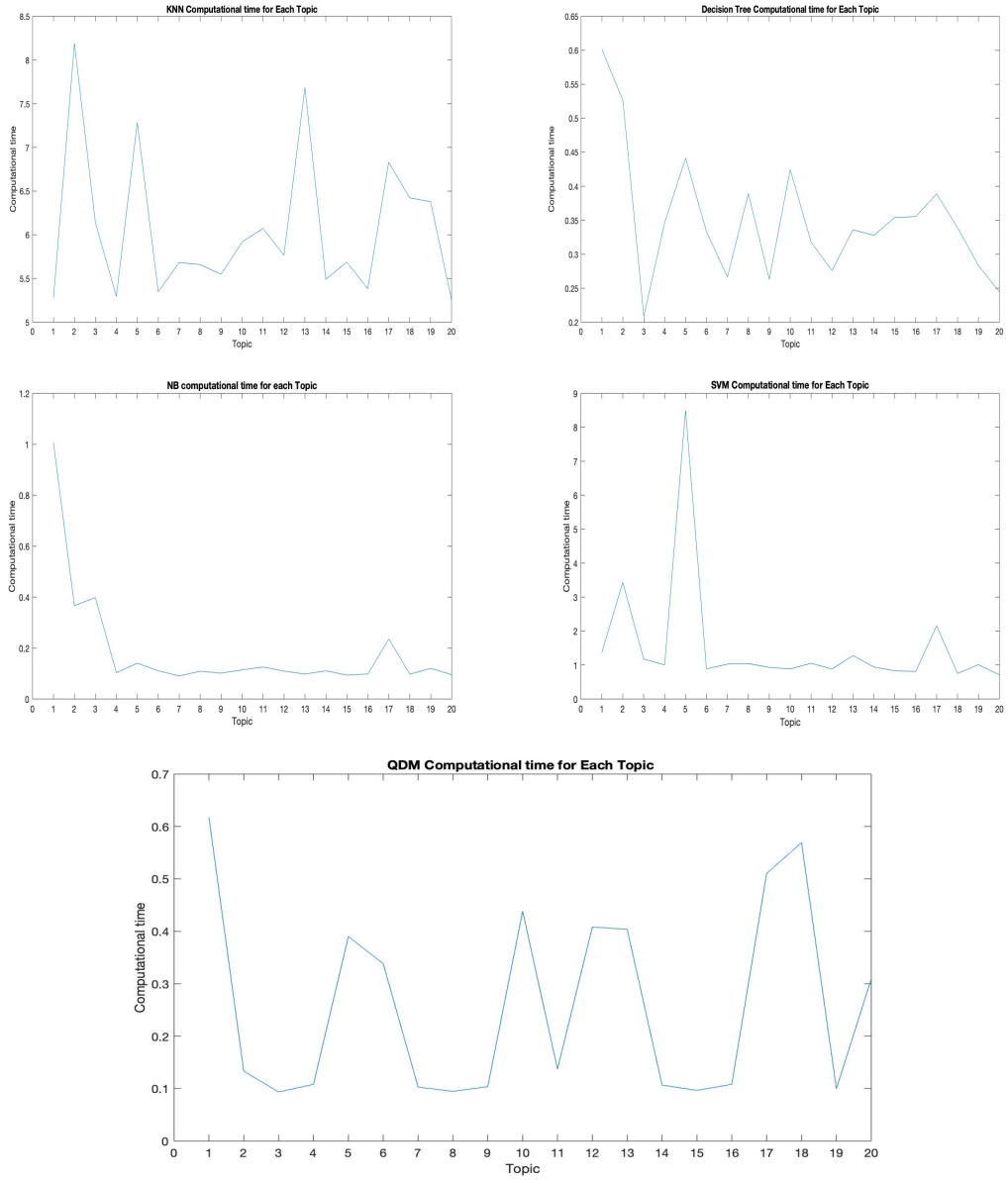


Figure 4.39: Computational time of KNN, DT, NB, SVM, and QDM on 20 Newsgroup Text Corpora for each topic with 100 features

improves with an increasing number of sampling as compared to KNN and SVM. QDM takes less time than the baselines in terms of computation for several topics, which is shown in Figures 4.40, 4.41, and 4.42.

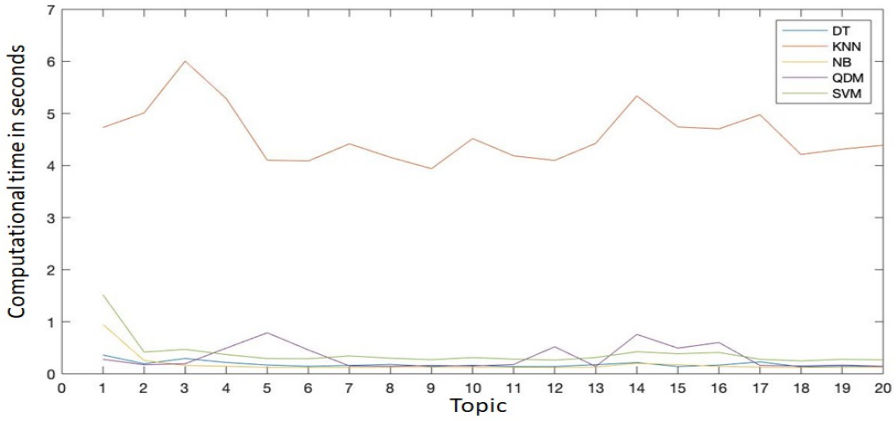


Figure 4.40: Computation time of QDM, SVM, NB, DT, and KNN with 30% training samples and rest for prediction

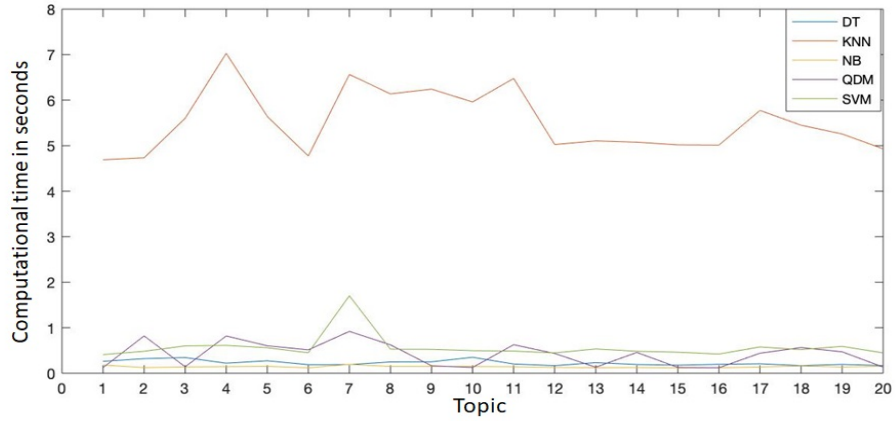


Figure 4.41: Computation time of QDM, SVM, NB, DT, and KNN with 40% training samples and rest for prediction

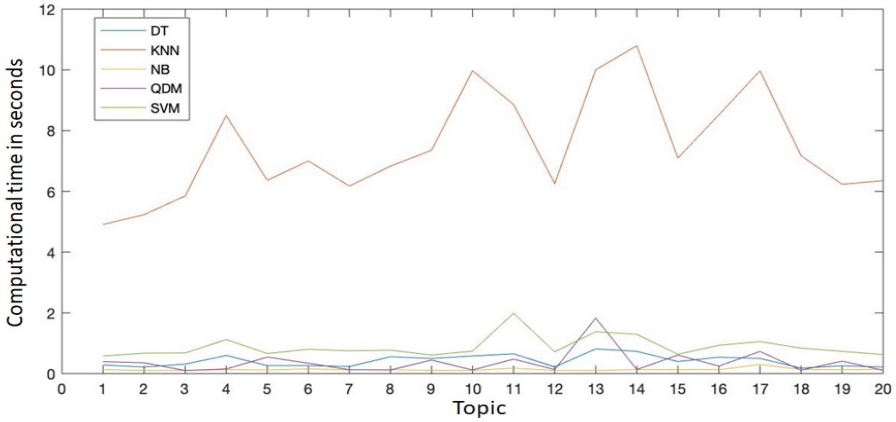


Figure 4.42: Computation time of QDM, SVM, NB, DT, and KNN with 50% training samples and rest for prediction

4.13 Case study on Failure and Success Analysis of QDM

It is essential to get some intuition behind the reason why QDM fails to perform for some categories and outperforms baselines for other categories. The experimental analysis has been done on a range of features and training samples in order to better understand the behaviour of the model. Some topics were selected where QDM failed to perform. For instance, if we consider Topic 20 from Section 4.7 on the 20Newsgroup, it can be seen that precision was lower than that of other baselines. QDM and other baselines were trained on a diverse range of training samples. Figures 4.12 and 4.15 shows that QDM precision was lower than that of the others, which is due to the fact that Topic 20 has a very low number of training samples in the whole training set. For example, 5% and 10% samples are considered to be very low training samples for this specific topic. However, when the percentage of training samples was increased, then the performance of Topic 20 started increasing, which can be seen in Figures 4.18, 4.21, 4.24, and 4.27. With increasing training samples, the top features also play an important role. It can also be seen that Topic 20 in particular worked quite well when the top 5, 10, and 15 features were selected, which can be seen in Figure 4.5 and Table 4.6. The features corresponding to Topic 20 had a high χ^2 for a few terms and a low χ^2 for others. Therefore, when those top features were selected, the precision was higher, but when the range of features started increasing with low value; then precision started decreasing due to low χ^2 values. QDM precision for Topic 19 was low in many cases due to low training samples, which can be seen in Figure 4.5. Topic 11 had a slightly higher number of training samples than other topics. When QDM was trained with fewer training samples, then recall and f-measure were lower, but performance started improving when the number of training samples started increasing, as can be seen in Figures 4.25, 4.26, 4.28, and 4.29. However, this improvement applies for some top ranges of selected features, i.e., top 30, 40, 50, 70, and 100. Generally speaking, QDM performance can be improved if a large number of training samples with high quality features is used for training. The autoencoder was also used in the place of the χ^2 feature selection model to check the performance, and it has also shown some promising results with different numbers of hidden sizes and training iterations. It is possible to train the autoencoder with higher dimensions and improve the proposed classifier performance, but the autoencoder is an unsupervised model where dependency is lost among the samples and target class; it is also time consuming to train the autoencoder. For those topics where QDM performance is not so effective in the fixed hyperparameter settings, it is possible to tune the hyperparameters and improve the performance for those *hard topics* as discussed in

Section 4.9. QDM was also compared with NN based classification models (see Section 4.11) and it work fairly well up to certain range of features by tuning different hyperparamaters, but a larger feature size is one limitation with the QDM. One advantage of using QDM for classification is that we have a degree of flexibility to tune the hyperparamaters and improve the performance of those categories (obtained results in Sections 4.9, and 4.11) whose performances were low in fixed settings of QDM (obtained results in Sections 4.5, 4.6, 4.7, 4.8, and 4.10).

4.14 Computational Complexity or Big $\mathcal{O}$ Analysis

In any new machine learning classifier, it is essential to check the computational complexity (Big $\mathcal{O}$), which is the main bottleneck. Please refer to Section 3.2.3.2 to understand this Big $\mathcal{O}$ analysis. Some notation needs to be mentioned here: N is the number of training samples, and k is the feature dimension. Sometimes N_{train} and N_{test} are used, which indicate the number of samples in training and the test set.

The training complexity of QDM can be seen in the following training steps:

- The first step is to estimate the probability p of samples in category c and the probability q of not being in the category. Element-wise multiplication was used for the computation, which was defined as $\mathcal{O}(N_{train}k)$ in training.
- Then, the eigendecomposition step was utilized, where $eigs(p'p - q'q, k-2)$ allows to obtain a diagonal matrix containing eigenvalues on the main diagonal, and another matrix whose columns are the corresponding eigenvectors. The computation complexity in this process was $\mathcal{O}(k^3)$

So, the whole time complexity in the training steps was $\mathcal{O}(N_{train}k + k^3)$. Furthermore, testing complexities from the test step $\langle x|\Delta|x\rangle$ were $\mathcal{O}(N_{test}k^3)$.

The computational complexities of the baselines are also essential to mention. The computational complexities of Naive Bayes in the training procedure were $\mathcal{O}(Nk)$, where the frequency of each feature value k_i is estimated for each class. In the testing step, run time complexity was $\mathcal{O}(ck)$, where each value of k needs to be retrieved for each class of c.

The training time complexity of SVM was $\mathcal{O}(N^3)$; however, SVM should be avoided if N is too large. The runtime complexity of SVM was $\mathcal{O}(s_v k)$, where s_v is the number of support vectors.

In the case of KNN, training time complexity was $\mathcal{O}(nNk)$, where n is the number of neighbors. It loops across each training sample and estimates the distance between training samples and new samples. Space complexity was $\mathcal{O}(Nk)$. Generally, training was faster in KNN, but the testing phase needs memory and time. It simply needs more time to scan the whole data-points, requiring more memory to store the training data.

The training time complexity of the decision tree was $\mathcal{O}(N \log Nk)$, and run-time complexity was $\mathcal{O}$(depth of the tree). The decision tree works fine with large data but low dimensionality.

CHAPTER 5

Discussion

The discussion of this work lies in the effectiveness and efficiency of the classification task. The main challenge in classification is how to obtain optimal effectiveness and efficiency despite complexity of data (RQ1, RQ2, RQ3, RQ4 mentioned in Section 1.3).

The classification model is proposed to tackle the challenges that arise due to a diverse range of features, training samples, and categories. The proposed QDM also has flexibility with hyperparamater tuning. It is inspired by quantum SDT with the aim of providing more effectual signals so that these signals are less susceptible to classification errors. The proposed classification model was validated on several datasets, including text and image datasets. The experimental results have shown that QDM performs well where classical models cannot perform. To check the effectiveness of the proposed model, analysis has been done on diverse ranges of features and training samples. QDM outperformed baselines for some categories but sometimes failed to do so for other categories in a normal setting. The obtained results led to the conclusion that the performance of QDM also depends upon the number of training samples, quality of features, and hyperparamater tuning. So if the training sample is very small, then performance is low, and performance starts improving with an increasing number of training samples. In addition, the features with high χ^2 values improve QDM performance. A case study has also been done on QDM, which can be found in Section 4.13. There are many hard categories where QDM performance is bit lower. However, it is possible to improve the performance of those hard topics by tuning the hyperparameter as discussed in Section 4.9.

The proposed model outperforms several baselines in terms of precision for several cases and has comparable results in others. QDM outperforms most of the baselines in terms of

recall in almost all situations when the image dataset is used. F-measure is also high for most of the cases and has comparable results in several cases.

The performance of the proposed model depends upon the selected hyperparamater λ and final detection boundary (to tackle RQ3). Precision, recall, and f-measure can be tuned based on the selected parameter values as discussed in Section 4.9. It is possible to observe in Sections 4.9 and 4.9.1 that QDM outperformed SVM for almost all the categories in terms of precision, recall, and f-measure when λ was tuned. Another hyperparamater is the final detection boundary which can be also tuned in order to improve the evaluation measure (refer to Sections 4.9 and 4.9.2). These parameters can be tuned for particular datasets, as shown in Section 4.9.

QDM is also compared with NN based classification models (see Section4.11) and it works fairly well up to some certain range of feature size. The hyperparamater tuning improved the performance of QDM to some degree. However, the performance of QDM starts deteriorating when the feature size is larger. It is essential to point out that the proposed QDM is not a neural model, but it is like a non-neural model (also known as a classical ML model) based on the statistics of training samples. We use some top features in non-neural models using feature selection methods. However, the working architecture of NN-based classification methods are different. We feed the whole feature to the NN-based classification model, and the NN model selects some high-level and low-level features automatically, which work like BlackBox. We also used a simple neural network autoencoder to reduce the feature's dimensionality as a feature reduction method in Section 4.10. The chi-square feature selection method is mostly used in this thesis (except Section 4.10) to select top features for the QDM along with baselines, but an autoencoder (which is also more expensive in terms of computational cost to use as a feature reduction method) is also used to select some features in Section 4.10. There are some advantages and disadvantages of neural and non-neural models. The main limitations of non-neural models are the large-sized datasets. Such non-neural models are effective on small and medium-sized datasets, but they become ineffective when dealing with large-sized datasets. Neural-based approaches have shown to be effective in handling large-sized datasets. However, NN-based models are computationally expensive, and they require GPU for training if we want results in a reasonable time because of the large-sized datasets and large number of paramaters. In contrast, non-neural models are cheap in terms of computational costs, and they perform better than neural models when it comes to small and medium-sized datasets. Non-neural models are easy to understand and interpret because they involve classical feature engineering, but this is one of the

biggest challenges with the neural models. It is more straightforward and flexible to tune hyperparameters in non-neural models because we have some understanding of data and models. However, neural models are often considered "black box", which is still a challenging task.

QDM works well in terms of high precision, recall, and f-measure based on proper hyperparameter tuning. QDM performance can be tuned based on the metric requirements because it provides more degrees of flexibility. For instance, there are many cases where high recall is needed, especially when output-sensitive predictions are required. For example, high recall is necessary for predicting terrorists or predicting cancer, as they must cover false negatives as well. More formally, it may be acceptable if a non-cancer tumor is labeled as cancerous, but a cancerous tumor cannot be labeled as non-cancerous because it would be more dangerous in this situation. It may be a crime in the case of rare cancer modeling where false-negatives are not considered. In most medical diagnostic problems, false-negatives are more dangerous than false-positives for early diagnosis. So, formally speaking, recall becomes a more important measurement in such a case because it considers false-negatives. Another example is the media monitoring system, where all the customers expect a high recall. This simply means that they never want to miss an article about the topics or domain they are interested in. It is not usually a problem when you get some noise in your article feed, so precision is not as important as recall in this situation. In all applications where you want to cover the false negatives, high recall is useful.

The proposed model is a binary classifier used to solve the problem of multiple class problems using a one-vs-all strategy. It is essential to mention that the multi-class and multi-label classification problems are generally solved using binary classifiers by decomposing them into binary classification problems [26, 3, 7] in traditional ML classification tasks.

Macro-average and micro-average analysis is also performed to check the overall effectiveness (to answer RQ2) of QDM. Whenever we use a one-vs-all strategy, the main downside is the class imbalance issue, which leads to low performance, so the micro-average metric is used. Macro-average is useful when we want to know the overall classifier performance across the different sets of data, and micro-average is useful when there is variation in dataset size, and we need to weigh each instance (i.e., documents) equally. Macro-average is not effective in this sense with the existing classification model when we average all the classes average contributions. However, QDM outperformed the baselines in terms of macro-average precision, as can be seen in Table 4.12 (RQ2). It is also possible to tune the hyperparameters and improve macro-average and micro-average results (as shown in Section 4.9).

The proposed model is also effective for high efficiency in terms of computational cost (RQ4 is the efficiency issue as mentioned in section 1.3), which can be seen in Section 4.12. Computational time is the amount of time that it takes to complete the operations for a given function. The proposed model takes much less time than SVM and KNN in terms of computation. The proposed model can be more useful than SVM and KNN because of its high efficiency and can be used safely where high efficiency is required. SVM takes a long time to compute because it is based on the kernel function. It also depends on the regularization parameter and data sizes (number of samples and features). On the other hand, KNN is a lazy algorithm, and it does not generalize the data in advance; besides, each time prediction is required, it scans the entire historical database. Generally, training is faster in KNN, but the testing phase needs memory and time. It needs more time to scan all the data-points, which requires more memory to store the training data. Of course, the computational cost of QDM also depends upon the number of features and samples. The training and prediction phases would be faster, with fewer data points. An increase in the data size leads to higher computational costs but also provides effectiveness in the model. The details of computational complexity or Big $\mathcal{O}$ Analysis with baselines are discussed in Section 4.14.

Like SVM, QDM also considers all training input vectors to construct the positive and negative state vectors, and then estimate the detectors Δ and $\Delta^{\perp}$ that best determine the two classes. We conjecture that the estimation procedure can better grasp the abstract patterns of input features as an SVD decomposition is used. In comparison, SVM is focused on the individual training samples and is prone to overfit. Some analogies can be seen among QDM, SVM, and NN. Generally, SVM uses distances to measure the closeness, while NN basically computes inner product to obtain class labels at the last layer. The density operator interacts with each document with Born's rule, which essentially computes squared inner product. So SDT is more analogous to NN in this way. By following the idea of SDT, the projections are just formulations of the hypotheses in SDT, which are the binary labels here in our case.

The proposed QDM for classification can be used safely in many applied ML domains. This model can be used for a number of classification tasks, i.e., the classification problem in NLP (sentiment analysis, topic detection, document or text or sentence classification [61], entity detection, token selection, relation classification [110], sarcasm detection, etc.), image classification tasks (face recognition, disease classification in the biomedical domain, etc.), sound classification, etc.

CHAPTER 6

Conclusion and Future Works

This thesis aims to tackle the challenges of classification effectiveness and efficiency due to data complexity (as mentioned in RQ1, RQ2, RQ3, RQ4). QDM classifier is proposed to deal with ineffectiveness due to the complexity of data. The proposed classification model is very flexible in hyperparameter tuning and improves performance metrics by proper tuning. The proposed model outperformed all the baselines in terms of precision, recall, f-measure, and accuracy in several cases. The proposed model outperformed baselines for many categories but is less effective in some cases. However, it is also possible to improve the performance of those less effective categories by proper hyperparameter tuning. To check the effectiveness of the proposed model, analysis has been done on several datasets. The proposed classification model's effectiveness depends on several criteria, such as the quality of features and training samples, the selected hyperparameter value, and others. It is possible to tune the hyperparameter of QDM and improve the evaluation metric for particular datasets. QDM is also compared with NN and showed effectiveness. It can be seen in the experiment section that QDM outperformed baselines when the hyperparameter was tuned. QDM also outperformed baselines when the macro-average metric is considered, which provides the overall effectiveness of the classifier.

The proposed model's efficiency is also good, and it takes less time for computation, which is another advantage of the proposed model. There are very limited works that use quantum formalism for classification tasks. This work could be the new direction to better understand quantum-inspired ML frameworks in the future. Further research is required at the theoretical and applied level in order to investigate in more detail. One research area, in particular, could be the investigation of the encoding step in order to find the most appropriate encod-

ing procedure to improve the classifier performance. The other area could be the use of other quantum states such as entangled states and squeezed states, which have shown potential for obtaining effective detection in communication systems.

References

[1] Abbey, C. K., and Eckstein, M. P. Classification image analysis: Estimation and statistical inference for two-alternative forced-choice experiments. *Journal of vision 2*, 1 (2002), 5–5.

[2] Abdi, H. Signal detection theory (sdt). *Encyclopedia of measurement and statistics* (2007), 886–889.

[3] Adnan, M. N., and Islam, M. Z. One-vs-all binarization technique in the context of random forest. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2015), pp. 385–390.

[4] Aggarwal, C. C. *Data classification: algorithms and applications*. CRC press, 2014.

[5] Aggarwal, C. C., and Zhai, C. A survey of text classification algorithms. In *Mining text data*. Springer, 2012, pp. 163–222.

[6] Ahumada, A. J. Classification image weights and internal noise level estimation. *Journal of Vision 2*, 1 (2002), 8–8.

[7] Aly, M. Survey on multiclass classification methods. *Neural Netw 19* (2005), 1–9.

[8] Asim, M. N., Khan, M. U. G., Malik, M. I., Dengel, A., and Ahmed, S. A robust hybrid approach for textual document classification. In *2019 International conference on document analysis and recognition (ICDAR)* (2019), IEEE, pp. 1390–1396.

[9] Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning* (2012), pp. 37–49.

[10] Banks, W. P. Signal detection theory and human memory. *Psychological bulletin 74*, 2 (1970), 81.

[11] Benenti, G., Casati, G., and Strini, G. *Principles of quantum computation and information: Volume II: Basic Tools and Special Topics*. World Scientific, 2007.

[12] Bhushan, S. B., and Danti, A. Classification of text documents based on score level fusion approach. *Pattern Recognition Letters 94* (2017), 118–126.

[13] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

[14] Cai, D., He, X., and Han, J. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering 17*, 12 (2005), 1624–1637.

[15] Camacho-Collados, J., and Pilehvar, M. T. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *arXiv preprint arXiv:1707.01780* (2017).

[16] Cariolaro, G. *Quantum communications*. Springer, 2015.

[17] Cariolaro, G. Quantum decision theory: Analysis and optimization. In *Quantum Communications*. Springer, 2015, pp. 183–249.

[18] ChandraShekar, B., and Shoba, G. Classification of documents using kohonen's self-organizing map. *International Journal of Computer Theory and Engineering 1*, 5 (2009), 610.

[19] Chatterjee, R., and Yu, T. Generalized coherent states, reproducing kernels, and quantum support vector machines. *arXiv preprint arXiv:1612.03713* (2016).

[20] Chen, J., Ji, S., Ceran, B., Li, Q., Wu, M., and Ye, J. Learning subspace kernels for classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), pp. 106–114.

[21] Chen, N., and Blostein, D. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR) 10*, 1 (2007), 1–16.

[22] Chernoff, H., and Moses, L. E. *Elementary decision theory*. Courier Corporation, 2012.

[23] Chinniyan, K., Gangadharan, S., and Sabanaikam, K. Semantic similarity based web document classification using support vector machine. *International Arab Journal of Information Technology (IAJIT) 14*, 3 (2017).

[24] Corrêa, R. F., and Ludermir, T. B. Automatic text categorization: case study. In *VII Brazilian Symposium on Neural Networks, 2002. SBRN 2002. Proceedings.* (2002), IEEE, p. 150.

[25] Dadgar, S. M. H., Araghi, M. S., and Farahani, M. M. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)* (2016), IEEE, pp. 112–116.

[26] de Carvalho, A. C., and Freitas, A. A. A tutorial on multi-label classification techniques. In *Foundations of computational intelligence volume 5*. Springer, 2009, pp. 177–195.

[27] Debole, F., and Sebastiani, F. An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and technology 56*, 6 (2005), 584–596.

[28] Di Buccio, E., Li, Q., Melucci, M., and Tiwari, P. Binary classification model inspired from quantum detection theory. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (2018), pp. 187–190.

[29] Djordjevic, I. *Quantum information processing and quantum error correction: an engineering approach*. Academic press, 2012.

[30] Eskin, E., Weston, J., Noble, W. S., and Leslie, C. S. Mismatch string kernels for svm protein classification. In *Advances in neural information processing systems* (2003), pp. 1441–1448.

[31] Feynman, R. P., Leighton, R. B., and Sands, M. Lectures on physics, vol. iii, 1965.

[32] Fisher, R. A. Statistical methods for research workers. In *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[33] Fung, G. M., Mangasarian, O. L., and Shavlik, J. W. Knowledge-based nonlinear kernel classifiers. In *Learning Theory and Kernel Machines*. Springer, 2003, pp. 102–113.

[34] Fung, G. M., Mangasarian, O. L., and Shavlik, J. W. Knowledge-based support vector machine classifiers. In *Advances in neural information processing systems* (2003), pp. 537–544.

[35] Fung, G. M., Mangasarian, O. L., and Smola, A. J. Minimal kernel classifiers. *Journal of Machine Learning Research 3*, Nov (2002), 303–321.

[36] Gao, P., Collins, L., Garber, P. M., Geng, N., and Carin, L. Classification of landmine-like metal targets using wideband electromagnetic induction. *IEEE Transactions on Geoscience and Remote Sensing 38*, 3 (2000), 1352–1361.

[37] Green, D. M., Swets, J. A., et al. *Signal detection theory and psychophysics*, vol. 1. Wiley New York, 1966.

[38] Greenwood, P. E., and Nikulin, M. S. *A guide to chi-squared testing*, vol. 280. John Wiley & Sons, 1996.

[39] Han, E.-H. S., and Karypis, G. Centroid-based document classification: Analysis and experimental results. In *European conference on principles of data mining and knowledge discovery* (2000), Springer, pp. 424–431.

[40] Han, X., Quan, L., Xiong, X., and Wu, B. Facing the classification of binary problems with a hybrid system based on quantum-inspired binary gravitational search algorithm and k-nn method. *Engineering Applications of Artificial Intelligence 26*, 10 (2013), 2424–2430.

[41] Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., and Gambetta, J. M. Supervised learning with quantum-enhanced feature spaces. *Nature 567*, 7747 (2019), 209.

[42] Haykin, S., and Thomson, D. J. Signal detection in a nonstationary environment reformulated as an adaptive pattern classification problem. *Proceedings of the IEEE 86*, 11 (1998), 2325–2344.

[43] He, L., Guo, C., Tiwari, P., Pandey, H. M., and Dang, W. Intelligent system for depression scale estimation with facial expressions and case study in industrial intelligence. *International Journal of Intelligent Systems* (2021), 1–18.

[44] Heisenberg, W. Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik. *Zeitschrift für Physik 43*, 3 (Mar 1927), 172–198.

[45] Helstrom, C. Quantum detection and estimation theory. *Academic press New York* (1963), 74–83.

[46] Helstrom, C. W. Detection theory and quantum mechanics. *Information and Control 10*, 3 (1967), 254–291.

[47] Helstrom, C. W. Quantum detection and estimation theory. *Journal of Statistical Physics 1*, 2 (1969), 231–252.

[48] Helstrom, C. W. *Statistical theory of signal detection: international series of monographs in electronics and instrumentation*, vol. 9. Elsevier, 2013.

[49] Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *The annals of statistics* (2008), 1171–1220.

[50] Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., and Guan, R. Text classification based on deep belief network and softmax regression. *Neural Computing and Applications 29*, 1 (2018), 61–70.

[51] Joachims, T. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Tech. rep., Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[52] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (1998), Springer, pp. 137–142.

[53] Kamavisdar, P., Saluja, S., and Agrawal, S. A survey on image classification approaches and techniques. *International Journal of Advanced Research in Computer and Communication Engineering 2*, 1 (2013), 1005–1009.

[54] Khamparia, A., Gupta, D., Nguyen, N. G., Khanna, A., Pandey, B., and Tiwari, P. Sound classification using convolutional neural network and tensor deep stacking network. *IEEE Access 7* (2019), 7717–7727.

[55] Kim, K., Akbar, I. A., Bae, K. K., Um, J.-S., Spooner, C. M., and Reed, J. H. Cyclostationary approaches to signal detection and classification in cognitive radio. In *2007 2nd ieee international symposium on new frontiers in dynamic spectrum access networks* (2007), IEEE, pp. 212–215.

[56] Kivinen, J., Smola, A. J., and Williamson, R. C. Online learning with kernels. *IEEE transactions on signal processing 52*, 8 (2004), 2165–2176.

[57] Koller, D., and Sahami, M. Toward optimal feature selection. Tech. rep., Stanford InfoLab, 1996.

[58] Korde, V., and Mahender, C. N. Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications 3*, 2 (2012), 85.

[59] Kyriakopoulou, A., and Kalamboukis, T. Using clustering to enhance text classification. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), pp. 805–806.

[60] Li, B., Bengtsson, T., and Bickel, P. Curse-of-dimensionality revisited: Collapse of importance sampling in very large scale systems. *Rapport technique 85* (2005), 205.

[61] Li, J., Hu, R., Liu, X., Tiwari, P., Pandey, H. M., Chen, W., Wang, B., Jin, Y., and Yang, K. A distant supervision method based on paradigmatic relations for learning word embeddings. *Neural Computing and Applications* (2019), 1–10.

[62] Liu, D. C., and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming 45*, 1-3 (1989), 503–528.

[63] Lo, H.-K., Spiller, T., and Popescu, S. *Introduction to quantum computation and information*. World Scientific, 1998.

[64] Lu, D., and Weng, Q. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing 28*, 5 (2007), 823–870.

[65] Melucci, M. A basis for information retrieval in context. *ACM Transactions on Information Systems (TOIS) 26*, 3 (2008), 1–41.

[66] Melucci, M. *Introduction to information retrieval and quantum mechanics*. Springer, 2015.

[67] Melucci, M. Relevance feedback algorithms inspired by quantum detection. *IEEE Transactions on Knowledge and Data Engineering 28*, 4 (2015), 1022–1034.

[68] Melucci, M., and Baeza-Yates, R. *Advanced topics in information retrieval*, vol. 33. Springer Science & Business Media, 2011.

[69] Melucci, M., et al. Foundations and trends® in information retrieval. *Foundations and Trends® in Information Retrieval 6*, 4-5 (2012), 257–405.

[70] Melucci, M., and van Rijsbergen, K. Quantum mechanics and information retrieval. In *Advanced topics in information retrieval*. Springer, 2011, pp. 125–155.

[71] Michalopoulou, Z.-H., Nolte, L. W., and Alexandrou, D. Performance evaluation of multilayer perceptrons in signal detection and classification. *IEEE transactions on neural networks 6*, 2 (1995), 381–386.

[72] Moreira, C., Tiwari, P., Pandey, H. M., Bruza, P., and Wichert, A. Quantum-like influence diagrams for decision-making. *Neural Networks 132* (2020), 190–210.

[73] Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[74] Murphy, K. P., et al. Naive bayes classifiers. *University of British Columbia 18* (2006).

[75] Murray, R. F. Classification images: A review. *Journal of vision 11*, 5 (2011), 2–2.

[76] Murray, R. F. Classification images in a very general decision model. *Vision Research 123* (2016), 26–32.

[77] Murray, R. F., Bennett, P. J., and Sekuler, A. B. Optimal methods for calculating classification images: Weighted sums. *Journal of Vision 2*, 1 (2002), 6–6.

[78] Nasios, N., and Bors, A. G. Kernel-based classification using quantum mechanics. *Pattern Recognition 40*, 3 (2007), 875–889.

[79] Neyman, J., and Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. R. Soc. Lond. A 231*, 694-706 (1933), 289–337.

[80] Neyman, J., and Pearson, E. S. The testing of statistical hypotheses in relation to probabilities a priori. In *Mathematical Proceedings of the Cambridge Philosophical Society* (1933), vol. 29, Cambridge University Press, pp. 492–510.

[81] Nikulin, M. Chi-squared test for normality. In *Proceedings of the International Vilnius Conference on Probability Theory and Mathematical Statistics* (1973), vol. 2, pp. 119–122.

[82] Phyu, T. N. Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (2009), vol. 1, pp. 18–20.

[83] Pritchett, L. M., and Murray, R. F. Classification images reveal decision variables and strategies in forced choice tasks. *Proceedings of the National Academy of Sciences 112*, 23 (2015), 7321–7326.

[84] Punera, K., Rajan, S., and Ghosh, J. Automatically learning document taxonomies for hierarchical classification. In *Special interest tracks and posters of the 14th international conference on World Wide Web* (2005), pp. 1010–1011.

[85] Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. Bgsa: binary gravitational search algorithm. *Natural Computing 9*, 3 (2010), 727–745.

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

[87] Rosipal, R., Trejo, L. J., and Matthews, B. Kernel pls-svc for linear and nonlinear classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003), pp. 640–647.

[88] Russell, S., Norvig, P., and Intelligence, A. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25*, 27 (1995), 79–80.

[89] Sammut, C., and Webb, G. I., Eds. *Leave-One-Out Cross-Validation*. Springer US, Boston, MA, 2010, pp. 600–601.

[90] Schölkopf, B., Smola, A. J., Bach, F., et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[91] Schuld, M., and Killoran, N. Quantum machine learning in feature hilbert spaces. *Physical review letters 122*, 4 (2019), 040504.

[92] Sebastiani, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR) 34*, 1 (2002), 1–47.

[93] Singh, U., and Hasan, S. Survey paper on document classification and classifiers. *Int. J. Comput. Sci. Trends Technol 3*, 2 (2015), 83–87.

[94] Swets, J. A. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Psychology Press, 2014.

[95] Swets, J. A., and Green, D. M. Signal detection by human observers. Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF ELECTRONICS, 1963.

[96] Swets, J. A., Tanner Jr, W. P., and Birdsall, T. G. Decision processes in perception. *Psychological review 68*, 5 (1961), 301.

[97] Tan, S., Cheng, X., Wang, B., Xu, H., Ghanem, M. M., and Guo, Y. Using dragpushing to refine centroid text classifiers. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 653–654.

[98] Tiwari, P., Dehdashti, S., Obeid, A. K., Melucci, M., and Bruza, P. Kernel method based on non-linear coherent state. *arXiv preprint arXiv:2007.07887* (2020).

[99] Tiwari, P., and Melucci, M. Multi-class classification model inspired by quantum detection theory. *arXiv preprint arXiv:1810.04491* (2018).

[100] Tiwari, P., and Melucci, M. Towards a quantum-inspired framework for binary classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018), pp. 1815–1818.

[101] Tiwari, P., and Melucci, M. Binary classifier inspired by quantum theory. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 10051–10052.

[102] Tiwari, P., and Melucci, M. Towards a quantum-inspired binary classifier. *IEEE Access 7* (2019), 42354–42372.

[103] Tiwari, P., Zhu, H., and Pandey, H. M. Dapath: Distance-aware knowledge graph reasoning based on deep reinforcement learning. *Neural Networks 135* (2021), 1–12.

[104] Trieu, L. Q., Tran, H. Q., and Tran, M.-T. News classification from social media using twitter-based doc2vec model and automatic query expansion. In *Proceedings of the Eighth International Symposium on Information and Communication Technology* (2017), pp. 460–467.

[105] Trunk, G. V. A problem of dimensionality: A simple example. *IEEE Transactions on pattern analysis and machine intelligence*, 3 (1979), 306–307.

[106] Uprety, S., Tiwari, P., Dehdashti, S., Fell, L., Song, D., Bruza, P., and Melucci, M. Quantum-like structure in multidimensional relevance judgements. In *European Conference on Information Retrieval* (2020), Springer, pp. 728–742.

[107] Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 2013.

[108] Von Neumann, J. *Mathematical foundations of quantum mechanics: New edition*. Princeton university press, 2018.

[109] Wang, D., Lu, C., Wu, J., Liu, H., Zhang, W., Zhuang, F., and Zhang, H. Softly associative transfer learning for cross-domain classification. *IEEE transactions on cybernetics 50*, 11 (2019), 4709–4721.

[110] Wang, D., Tiwari, P., Garg, S., Zhu, H., and Bruza, P. Structural block driven enhanced convolutional neural representation for relation extraction. *Applied Soft Computing 86* (2020), 105913.

[111] Wang, W., and Yu, B. Text categorization based on combination of modified back propagation neural network and latent semantic analysis. *Neural computing and applications 18*, 8 (2009), 875.

[112] Wetzel, S. J. Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Physical Review E 96*, 2 (2017), 022140.

[113] Wixted, J. T. Dual-process theory and signal-detection theory of recognition memory. *Psychological review 114*, 1 (2007), 152.

[114] Wu, H., Phang, T. H., Liu, B., and Li, X. A refinement approach to handling model misfit in text categorization. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), pp. 207–216.

[115] Wu, J., Yang, Y., Lei, Z., Wang, J., Li, S. Z., Tiwari, P., and Pandey, H. M. An end-to-end exemplar association for unsupervised person re-identification. *Neural Networks 129* (2020), 43–54.

[116] Xie, Q., Tiwari, P., Gupta, D., Huang, J., and Peng, M. Neural variational sparse topic model for sparse explainable text representation. *Information Processing & Management 58*, 5 (2021), 102614.

[117] Xu, S., Li, Y., and Wang, Z. Bayesian multinomial naïve bayes classifier to text classification. In *Advanced multimedia and ubiquitous engineering*. Springer, 2017, pp. 347–352.

[118] Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., and Ma, W.-Y. Ocfs: optimal orthogonal centroid feature selection for text categorization. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 122–129.

[119] Yang, Y. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (1995), pp. 256–263.

[120] Yang, Y. An evaluation of statistical approaches to text categorization. *Information retrieval 1*, 1-2 (1999), 69–90.

[121] Yang, Y., and Pedersen, J. O. A comparative study on feature selection in text categorization. In *Icml* (1997), vol. 97, Nashville, TN, USA, p. 35.

[122] Yang, Y., Zhang, T., Cheng, J., Hou, Z., Tiwari, P., Pandey, H. M., et al. Cross-modality paired-images generation and augmentation for rgb-infrared person re-identification. *Neural Networks 128* (2020), 294–304.

[123] Yao, L., Mao, C., and Luo, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 7370–7377.

[124] Yu, B., Xu, Z.-b., and Li, C.-h. Latent semantic analysis for text categorization using neural network. *Knowledge-Based Systems 21*, 8 (2008), 900–904.

[125] Zdrojewska, A., Dutkiewicz, J., Jędrzejek, C., and Olejnik, M. Comparison of the novel classification methods on the reuters-21578 corpus. In *International Conference on Multimedia and Network Information System* (2018), Springer, pp. 290–299.

[126] Zeng, H.-J., Wang, X.-H., Chen, Z., Lu, H., and Ma, W.-Y. Cbc: Clustering based text classification requiring minimal labeled data. In *Third IEEE International Conference on Data Mining* (2003), IEEE, pp. 443–450.

[127] Zhang, Y., Liu, Y., Li, Q., Tiwari, P., Wang, B., Li, Y., Pandey, H. M., Zhang, P., and Song, D. Cfn: A complex-valued fuzzy network for sarcasm detection in conversations. *IEEE Transactions on Fuzzy Systems* (2021), 1–15.

[128] Zhang, Y., Tiwari, P., Song, D., Mao, X., Wang, P., Li, X., and Pandey, H. M. Learning interaction dynamics with an interactive lstm for conversational sentiment analysis. *Neural Networks 133* (2021), 40–56.

[129] Zheng, W., Qian, Y., and Lu, H. Text categorization based on regularization extreme learning machine. *Neural Computing and Applications 22*, 3-4 (2013), 447–456.

Acknowledgments

This reserach has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 721321.