University of Bahrain

College of Science

Department of
Postgraduate Programs

A Novel Incremental Deep Learning Technique for QoS Interpretation and QoE Prediction for Top Social Media Platforms

Thesis submitted in partial fulfilment of the requirements for the degree of M.Sc. in Big Data Science and Analytics

Prepared by:

Isa Yaser Isa Ajlan AlHeddi

202100022

Supervised by:

Dr. Riadh Bin Mohammad Ksantini

Associate Professor

University of Bahrain

Kingdom of Bahrain

March 2023

University of Bahrain

College of Science

Department of Postgraduate Programs



A Novel Incremental Deep Learning Technique for QoS Interpretation and QoE Prediction for Top Social Media Platforms

Thesis submitted in partial fulfilment of the requirements for the degree of M.Sc. in Big Data Science and Analytics

Prepared by:

Isa Yaser Isa Ajlan AlHeddi

202100022

Supervised by:

Dr. Riadh Bin Mohammad Ksantini

Associate Professor

University of Bahrain

Kingdom of Bahrain

March 2023

Defense Committee's Approval

University of Bahrain
Deanship of Graduate Studies
And scientific Research

جامعة البحرين
عمادة الدراسات العليا والبحث العلمي

Thesis Discussion Committee Decision

Student Name: Isa Yaser Isa Ajlan AlHeddi	Student ID: 202100022
Department: Choose an item.	College: **Science**
Program: **M.Sc. in Big Data Science and Analytics**	
Thesis Title: A Novel Incremental Deep Learning Technique for QoS Interpretation and QoE Prediction for Top Social Media Platforms	
Supervisor Name: **Dr. Riadh Ksantini**	Student GPA: 3.89

The Discussion Committee met on 22-May-23 In S41-1002. Upon discussing the student whose details are stated above in public, the discussion committee recommends the following:

The student is awarded a grade Choose an item. in the thesis subject.

☐ Without making any changes to the thesis.

☒ After making the modifications attached with this decision.

Discussion Committee Members Approval:

Name		Signature
1. **Dr. Riadh Ksantini**	(Advisor)	
2. **Prof. Nabil Hewahi**	(Internal Examiner)	Nabil Hewahi
3. **Prof. Ahmed Fawzi Otoom**	(External Examiner)	

6 Approved at the University Council 6th meeting, 2013
In accordance with Decision No. 736/2013 dated 14/3/2013
Modified by University Council Decision No. 2043/2013 dated 20/11/2013.

Abstract

Telecom Service Providers (TSP) face challenges in predicting Quality of Experience (QoE) as they strive to meet customer demands. Machine Learning (ML) has become a powerful tool for predicting QoE and helping TSPs exceed customer expectations. However, most existing ML algorithms for QoE prediction are batch learning-based, lacking real-time capabilities. To address this issue, a research study proposes an Incremental Discriminant based Support Vector Machine (IDSVM) model.

The IDSVM model combines global and local variants using a maximum margin-based and discriminant-based approach. Unlike traditional methods, it utilizes the input space instead of the feature space to generate the variance-covariance matrix. This incremental classifier is well-suited for real-time QoE prediction, aligning with the needs of TSPs.

To train the IDSVM model, the researchers utilized a dataset obtained from the Bahraini telecom regulator. This dataset, the first of its kind in deploying the novel classifier, contained technical data on the webpage accessibility of Facebook, Twitter, and Instagram. The dataset underwent several preparation phases, including data cleansing, output insertion, output balancing, and attribute selection using an autoencoder. The autoencoder generated deep features from the latent space, which served as inputs to the IDSVM model.

By developing the IDSVM model and leveraging the unique dataset, this research aims to improve QoE prediction in the telecom industry, enabling TSPs to enhance customer satisfaction and maintain a positive brand image. The evaluation of the novel classifier was done using multiple performance measures, including F1 score, false positive rate, and the area under the receiver operating characteristic (ROC) curve. The training time was also considered in the evaluation. Considering the ROC measure, IDSVM outperformed the other classifiers in Twitter (0.6744) and Instagram (0.6169) datasets and scored the second lowest training time (14.66 seconds).

In general, IDSVM model performance is marginally good, and it is considered a fast classifier. Therefore, it is well suited for the real-time applications.

Dedication

This thesis work is dedicated to whom have been the source of motivation and support.

To my **parents**, **wife**, and **siblings**.

Acknowledgements

Prophet Mohammed, may Allah bless him and grant him peace, said, "He who does not thank people does not thank Allah."

Hence, I would like to thank the University of Bahrain and Liverpool John Moores University for establishing the Big Data Science and Analytics master's program.

Special acknowledgment goes to:

Dr. Riadh Ksantini for his support and guidance during the thesis preparation.

Dr. Sawsan Hilal for her seamless dedication and support throughout the program.

Prof. Nabil Hewahi, **Dr. Hesham Al-Ammal**, **Dr. Ahmed Zeki** and **Dr. Abdulla AlQaddoumi** for their appreciated efforts in delivering the program courses.

Last but not least, I am thankful to the **Telecommunication Regulatory Authority in the Kingdom of Bahrain** for their support in providing the necessary data to conduct this research.

Finally, I would like to express my gratefulness to **Aysha**, **Hamad**, **Ahmad**, and **Saad** for their support and encouragement to pursue this program.

Table of Contents

Defense Committee's Approval ..... A

Abstract ..... B

Dedication C

Acknowledgements ..... D

Table of Contents ..... E

List of Tables ..... H

List of Figures I

List of Abbreviations ..... J

Chapter One Introduction to the Research 14

1.1. Overview and Background 14

1.2. Research Problem 16

1.3. Research Aim and Objectives 16

1.4. Research Significance 17

1.5. Research Limitations 18

1.6. Roadmap 18

Chapter Two Literature Review and Related Work 21

2.1. QoS and QoE in Telecom 21

2.1.1. Background 21

2.1.2. Importance of measuring QoE 22

2.2. QoE Inference 22

2.2.1. Subjective Method 23

2.2.2. Objective Method 24

2.2.3. Pseudo Subjective Quality Assessment 25

2.3. Statistical, Machine Learning and Deep Learning Algorithms in QoE 26
2.3.1. Statistical Algorithms 26
2.3.2. Machine Learning Models 28
2.3.3. Deep Learning Models 30
2.4. Incremental Learning 34
2.5. Novelty ..... **Error! Bookmark not defined.**
Chapter Three Methods and Procedures 40
3.1. Methods 40
3.1.1. IDSVM 41
3.2. Procedures 49
3.2.1. Data Cleansing 49
3.2.2. Output Insertion 51
3.2.3. Dimensionality Reduction 54
Chapter Four Experiments, Results and Discussion 59
4.1. Dataset Description 59
4.2. Experiment Settings and Protocols 60
4.2.1. Autoencoder Hyperparameters 61
4.2.2. DSVM Hyperparameters 64
4.3. Comparative Algorithms 67
4.3.1. KSVM 67
4.3.2. IKSVM 69
4.3.3. IKNSVM 69
4.4. Performance Measures 70

4.5. Results and Discussion 73
4.5.1. Full datasets 74
4.5.2. Redacted datasets.....81
Chapter Five Conclusion and Recommendation.....86
5.1. Conclusion 86
5.2. Recommendation 87
References.....88
الملخص 96

List of Tables

Table (2.1): MOS scale interpretation 23
Table (2.2): Potential PSNR to MOS translation.....25
Table (2.3): Strengths and weaknesses of the algorithms.....33
Table (2.4): Overview of studies on QoE prediction using ML and DL algorithms.36
Table (3.1): Threshold per attribute.....51
Table (4.1): Attribute category description.....60
Table (4.2): AE hyperparameters for top performance.....63
Table (4.3): IDSVM hyperparameters for top performance.....66
Table (4.4): Hyperparameters for compared algorithms.....67
Table (4.5): AE architectures description.....74
Table (4.6): AE performance.....75
Table (4.7): IDSVM performance.....75
Table (4.8): IDSVM performance for Facebook dataset.....76
Table (4.9): IDSVM performance for Twitter dataset.....77
Table (4.10): IDSVM performance for Instagram dataset.....78
Table (4.11): Training time per social media platform.....81
Table (4.12): Performance per classifier per dataset.....82
Table (4.13): Training time per classifier per dataset.....83

List of Figures

Fig. 3.1: Compound novel model.....40

Fig. 3.2: Between-class and within-class graphical illustration.....42

Fig. 3.3: Non-linear boundary and near-global support vectors in SVM. Adapted from (Burkov, 2019).....43

Fig. 3.4: Depiction of the subsets (a) ***D***, (b) ***R***, (c) ***M*** as given by the KTT. Adapted from (Soula et al., 2020).....47

Fig. 3.5: Data preparation procedure flowchart.....49

Fig. 3.6: Illustrative AE Architecture. Adapted from (Flores, 2019).....56

Fig. 3.7: Steps followed for data preparation and model building.....57

Fig. 4.1: IDSVM performance for architecture 1.....80

List of Abbreviations

3G	Third Generation of Mobile Networks
4G	Fourth Generation of Mobile Networks
5G	Fifth Generation of Mobile Networks
ADASYN	Adaptive Synthetic Sampling
ADT	Adaboost Decision Tree
AE	Autoencoder
ANFIS	Adaptive Neuro Fuzzy Inference System
BN	Bayesian Network
BPNN	Back Propagation Neural Network
CNN	Convolutional Neural Networks
CTV	Cumulative Traffic Volume
dBm	Decibel Milliwatts
DL	Deep Learning
DSVM	Discriminant based Support Vector Machine
DT	Decision Tree
GRU	Gated Recurrent Unit
IDSVM	Incremental Discriminant based Support Vector Machine
IKNSVM	Incremental Kernelized Non-Parametric Support Vector Machine
IKSVM	Incremental Kernelized Support Vector Machine
IPTV	Internet Protocol Television
ISVM	Incremental Support Vector Machine
ITU	International Telecommunications Union
JSON	JavaScript Object Notation
KKT	Karush–Kuhn–Tucker Conditions
KNDA	Kernelized Non-Parametric Discriminant Analysis
KNN	K-Nearest Neighbour

KSVM	Kernelized Support Vector Machine
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
Mbps	Megabit per Second
ML	Machine Learning
MLP	Multi-layer Perceptron
MLR	Multiple Linear Regression
MOS	Mean Opinion Score
MSE	Mean Squared Error
NB	Naïve Bayes
ANN	Artificial Neural Networks
PSNR	Peak Signal to Noise Ratio
PSQA	Pseudo Subjective Quality Assessment
QoE	Quality of Experience
QoS	Quality of Service
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
RSSI	Received Signal Strength Indicator
SISVM	Stacked Incremental Support Vector Machine
SMOTE	Synthetic Minority Over-sampling Technique
SSIM	Structural Similarity Index Measure
SVD	Singular Value Decomposition
SVM	Support Vector Machine

TCP	Transmission Control Protocol
TSP	Telecom Services Provides
TTFB	Time to First Byte
VoD	Video on Demand
VoIP	Voice over Internet Protocol

Chapter One

Introduction to the Research

1.1. Overview and Background

1.2. Research Problem

1.3. Research Aim and Objectives

1.4. Research Significance

1.5. Research Limitations

1.6. Roadmap

Chapter One
Introduction to the Research

1.1. Overview and Background

In this unprecedented era of advanced telecommunications, where the data exchange became the backbone of any economy, Telecom Services Provides (TSP) race towards providing the best services to the consumers. Nevertheless, the TSP are blind when it comes to the Quality of Experience (QoE) that is perceived by the consumers, unless explicitly declared by the consumers through a survey or so, or certain standard/criteria is followed. Moreover, the TSP require instant (real-time) prediction of the perceived QoE to take the necessary corrective measures in a timely manner (Diaz-Aviles et al., 2015).

According to this research, Efazati et al. (2020), the evaluation of the Quality of Service (QoS) parameters usually works by setting thresholds that rank/categorize the overall QoS parameters (e.g., average upload/download speeds, latency, jitter, packet discard, time to first byte, etc.). Precise prediction of the consumers' experience is essential for the TSP to stay competitive. Also, Diaz-Aviles et al. (2015) added that TSP ensure collecting huge amount of data from their networks and consumers to analyse the same and produce useful insights that will help them enhance their networks and services.

This research aims at experimenting deep learning technique that is complemented by novel classifier that uses incremental learning to predict the perceived QoE based on number of provided parameters that are related to QoS. It also aims at comparing the chosen algorithm with other algorithms to prove its effectiveness. The experience of accessing

homepage, login page and public social profile page will be predicted for the top social media platforms namely, Facebook, Twitter, and Instagram.

The research is conducted using data from the mobile crowdsourcing database of the Telecommunications Regulatory Authority in the Kingdom of Bahrain (Telecommunication Regulatory Authority, n.d.). The database lacked the output (i.e., the equivalent perceived QoE). The output was added following industry's best practices of interpreting the QoS parameters. Google Colab was used for plugging the QoE attribute (the output), data cleansing and for building the Autoencoder (AE) algorithm. As for the novel Incremental Discriminant based Support Vector Machine (IDSVM) algorithm, it was built using MATLAB.

The AE was trained using the database in question via data feedforward yielding to dimension reduction. The novel IDSVM algorithm was fed by the AE's output. The IDSVM algorithm was then used to classify the records into two classes, acceptable experience and inacceptable experience. This algorithm has the advantage of near-global data spread to improve classification. Moreover, this algorithm adopts the incremental learning approach.

In this research, a compound Machine Learning (ML) model was built that is able to classify the data with acceptable level of accuracy. With this level of performance, it is anticipated that this model could set the basis for real life applications in the telecommunications sector. The compound ML model is based on the work done in a paper that was used in the medical imaging field (Salhi et al., 2022). Moreover, this research is envisaged to add value to the literature related to area of predicting QoE using ML techniques. In addition, it is anticipated to prove the efficiency of incremental learning as opposed to batch learning.

1.2. Research Problem

In the telecom industry, satisfying and retaining consumers are becoming harder than ever (Clark, 2020). To meet the consumers' expectations, it is essential to understand how the provided service is perceived. It simply tells the TSP how good or bad the consumers think the provided telecom service is. Nevertheless, determining the perceived QoE in real-time with high level of accuracy remains a challenge for the TSP.

Engaging the consumers directly through surveys to provide feedback about how satisfied they are with the level of the offered telecom services is a costly, time-consuming process and infeasible for large scale settings (Diaz-Aviles et al., 2015). Hence, quick, cost efficient and effective solutions had to be developed to fill the gap of understanding the QoE from the consumers' perspective without a direct engagement. The idea for this research problem has emerged from the challenge explained above.

1.3. Research Aim and Objectives

The aim of this research is to incorporate ML algorithms into real life problem to prove its effectiveness in resolving such problems. More specifically, the objective is to pave the way for the ML algorithms to be integral part of the QoE prediction challenge after demonstrating its effectiveness on real cases. This could potentially set the scene for new era of cost-effective solutions for QoE prediction.

This research also aims for understanding the performance of the novel IDSVM algorithm and other algorithms in the literature that were used in resolving problems related to predicting the QoE. Furthermore, this research intended to experiment the efficiency of incremental learning in comparison to batch learning technique. Additionally, this research is done

with an objective of enriching the relevant literature with real-life based experiment.

The guiding research questions are as follows:

1. What challenges are faced by TSP with regard to predicting the perceived QoE?
2. What is the suitable ML algorithm for such challenges?
3. How effective is DL technique and discriminant-based Support Vector Machine (SVM) classifier in accurately predicting the perceived QoE?
4. How effective is the incremental learning in comparison to the batch learning?
5. How does the novel IDSVM perform in comparison to similar classifiers?

1.4. Research Significance

The use of the novel IDSVM algorithm was not considered in tackling telecom QoE prediction related issues, hence it can assist the TSPs to instantly predict the QoE. Moreover, this is the first time that such dataset is made available for research and experiment purposes, hence, experimenting this dataset with a novel model will add value to this research in terms of significance. This novel model is used in conjunction with the AE algorithm to make use of the dimensionality reduction power of the later.

The speed and the volume at which the data is generated by the telecom networks dictates the usage of advanced models to classify the perceived quality of experience. This necessitates building a model that is both a) able to reduce the data dimension to only retain the features that are

most relevant to the output and b) able to learn incrementally as batch learning is computationally costly. These challenges are addressed in this research through the studied model.

1.5. Research Limitations

In this research, only one dataset is used for the experimentations which has been collected by the researcher. The dataset was complex and required extensive data preparation work. The output (QoE) had to be synthetically produced as it was not part of the extracted dataset. The QoE was produced following industry's best practices. In addition, the studies that addressed the QoS interpretation and QoE prediction through ML and DL algorithms are relatively limited in the literature. Furthermore, literature about incremental ML algorithms is also limited. This is even more scarce when it comes to the papers that are concerned about the usage of incremental ML algorithms for QoE predictions. Moreover, another limitation is the platform to build the IDSVM algorithm, as not all programming languages and libraries support this algorithm, nevertheless this can be found predefined in the MATLAB version R2022a.

1.6. Roadmap

Apart from Chapter 1, this research contains the following chapters:

Chapter 2: In this chapter a background was provided about QoS and QoE in the telecom world along with the QoE inference methods. Also related works were presented, papers and research that utilized the ML methods to predict the QoE.

Chapter 3: This chapter includes thorough explanation of the novel classifier as well as the dataset preparation steps that took place.

Chapter 4: Results using multiple performance measures are presented in this chapter, for both, the novel classifier, and the other compared classifiers.

Chapter 5: This is the last chapter. Conclusion and recommendation for future work are given in this chapter.

Chapter Two

Literature Review and Related Work

2.1. QoS and QoE in Telecom

2.2. QoE Inference

2.3. Statistical, Machine Learning and Deep Learning Algorithms in QoE

2.4. Incremental Learning

2.5. Summary

Chapter Two

Literature Review and Related Work

2.1. QoS and QoE in Telecom

As per the research done by Fiedler et al. (2010), users' perception, experience, and expectations are linked to application and telecom network performance via QoE, which is often described by QoS parameters. The International Telecommunications Union (ITU) defines the QoS in its recommendation Telecommunications Standardization Sector of ITU (2008) as collective effect of service performance which determines the degree of satisfaction of a user of the service, while it defines the QoE as the overall acceptability of an application or service, as perceived subjectively by the end-user in its recommendation (Telecommunications Standardization Sector of ITU, 2019).

2.1.1. Background

The vast majority of the governments across the globe have taken the initiative to liberalize the telecom market to break the dominance of national telecom operators. Legislations are formulated to lift the entry barriers for potential investors that are interested in building additional telecom networks and eventually taking a share of the telecom market. By introducing the liberalization, the incumbent operator is no longer the sole provider of the telecom services. Such liberalization move has naturally forced the TSP to re-design their business models to constantly address consumers' demands and expectations (International Chamber of Commerce, 2007).

With the evolvement of the different applications and the diversity of the use cases of the telecom services, a recent paper by Gatara and Mzyece (2022) indicated that a transformation is taking place from QoS oriented mobile networks to QoE centric mobile networks.

2.1.2. Importance of measuring QoE

Consumers can use both standard and advanced devices to access mobile network services anywhere within the mobile networks' coverage region. Typically, customers have some expectations for the services offered, hence, maintaining satisfied consumers is vital to have profitable business. Therefore, TSP are sparing no efforts in applying state of the art techniques and methods to predict the perceived QoE via analysing the QoS parameters. The results of the analyses are then translated into action plans with the aim of maintaining or increasing the customer base. In a paper prepared by Ullah et al. (2019), it was added that TSP prefer to retain existing customer over attracting new ones. This is simply because attracting new customer costs five to six times more than retaining existing one. Unlike the prepaid customers, post-paid customers are less likely to churn. Customers would usually leave their TSP in seek for better QoE and/or better price. By increasing the customer retention rate by 1%, operator's share price may increase by 5%.

2.2. QoE Inference

QoE is conjugated to the consumer's perception of the provided telecom service. Consequently, they are directly proportional. The subjective nature of the QoE makes it a challenging parameter to be measured and quantified. Therefore, telecom professionals had to come up with methods that calculate this parameter. Throughout the studied

literature, three methods were identified that have been adopted to predict the perceived QoE as explained below. Studying such methods helped in producing the output for the examined dataset in this research.

2.2.1. Subjective Method

As the name suggests, this method lacks the logical/mathematical basis for the evaluation. ITU issued a recommendation Telecommunications Standardization Sector of ITU (2016), that pointed out that the subjective assessment relies on human rating of their perception of the quality of the telecom service under evaluation. Nevertheless, as per the research conducted by Piamrat et al. (2009), it is considered the most accurate method to judge on the quality of the provided telecom service, i.e., the perceived QoE, because there is no better indicator of the telecom service quality than the consumer.

The evaluation is based on a measure that is referred to as Mean Opinion Score (MOS). It is a common scale that is used to measure the quality of the provided telecom service. The way it works is that a user will be asked to rate the level of quality/satisfaction of a certain telecom service like web browsing, audio/speech quality or video quality. (Telecommunications Standardization Sector of ITU, 2016). Table 2.1 illustrates the MOS scale and the interpretation of each given score.

Table (2.1): MOS scale interpretation. Adapted from (Zhang & Wang, 2018)

MOS Score	Quality Level	Impairment Level
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Despite being the most accurate method, the research done by Piamrat et al. (2009) touched upon its drawbacks. This method has number of drawbacks that hinders its practical adoption/implementation. This method is considered expensive and time consuming as it involves recruiting manpower to run the evaluation. Moreover, it is hardly scalable, and it cannot be used as a tool for real-time observation.

The use cases of the subjective method are numerous. A closely relevant example is an experiment done by Da Hora et al. (2018) that gathered 8,689 web-browsing sessions to be rated by 241 volunteers in accordance with MOS scale. The authors of this paper have utilized the gathered ratings to set the basis for an objective method that is established on human ratings. Next, the objective method will be explained.

2.2.2. Objective Method

As mentioned in the previous section, the subjective method is impractical. Additionally, in a paper by Piamrat et al. (2009), it was mentioned that researchers looked for alternative methods that are both cost and time efficient and can be automated. The alternative methods are based on mathematical formulas and telecom networks performance parameters to produce a certain value that can be interpreted to binary/scaled evaluation scheme. There must be some sort of heuristic mapping between the objective results and the MOS subjective method to make sense out of the produced values. Such mapping is done using direct human ratings.

As an illustrative case of the objective method, Piamrat et al. (2009) has chosen to evaluate the perceived video quality using the proportion of a signal's maximal strength to the power of corrupting noise that compromises the accuracy of its representation, commonly referred to as Peak Signal to Noise Ratio (PSNR). It is basically calculated as the

difference in the Mean Squared Error (MSE) between the original frame and the distorted one. Table 2.2 below demonstrates how is the PSNR level in dB corresponds to the MOS value.

Table (2.2): Potential PSNR to MOS translation. Adapted from (Piamrat et al., 2009)

PSNR (dB)	MOS
> 37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
< 20	1 (Bad)

Another example for the adoption of the objective method, is research conducted by Da Hora et al. (2018), where QoE inference of web-browsing quality was based on three mathematical functions to map nine QoS metrices to user's experience.

From a webpage published by the International Telecommunications Union (2022), it can be deduced that the objective method is cheaper than the subjective method due to the limited required human intervention. Moreover, it is time efficient approach, systematic and highly scalable. Nevertheless, in this method not all factors/parameters can be quantified and taken into the consideration.

2.2.3. Pseudo Subjective Quality Assessment

One may argue that the Pseudo Subjective Quality Assessment (PSQA) method is similar to the objective method. However, Rubino and Varela (2004) in their research paper defined it as a method that simulate how the typical person would rate a telecom service using a statistical estimator. In contrast to other evaluation methods, which occasionally

depend on models of human perception to carry out the evaluation, PSQA creates a mapping between specific corresponding elements (e.g., quality-affecting network parameters and the perceived QoE). It learns the relationship between those parameters and perceived QoE in this way.

PSQA method presents advantages that are not in the first two methods. PSQA can provide real-time analysis and to show how each network parameter influences the perceived QoE (Piamrat et al., 2009).

The applications of the PSQA method are unlimited. For instance, a paper conducted by Shan et al. (2018) described a novel technique for gauging speech quality that does not rely on reference speech. They assessed speech quality using DL algorithm, namely Back Propagation Neural Network (BPNN). This is done using a dataset with a variety of manipulated speech signals and the corresponding original speech signal. Such dataset is used to train the algorithm. This allowed the algorithm to convert the manipulated speech signals to their corresponding original speech signals and accordingly rate the speech's quality. The achieved Root Mean Squared Error (RMSE) was 0.4612, which was the lowest among the tested methods (P.563, GMM-based and Autoencoder-based).

2.3. Statistical, Machine Learning and Deep Learning Algorithms in QoE

Statistical algorithms, ML algorithms, and DL algorithms are all types of algorithms used in data analysis and modelling, but they differ in their complexity and the types of problems they are best suited to solve.

2.3.1. Statistical Algorithms

Statistical algorithms are a type of mathematical models that use statistical techniques to make predictions or estimate the probability of

certain outcomes. They are often used for tasks such as hypothesis testing, estimation, and forecasting. There are many different types of statistical algorithms, including but not limited to linear algorithms, generalized linear algorithms, and time series algorithms. Linear algorithms, such as linear regression, are used to model the relationship between a dependent variable and one or more independent variables. Generalized linear algorithms are a generalization of linear algorithms that allow for nonlinear relationships. Time series algorithms are used to model time-dependent data, such as stock prices or weather patterns. Overall, statistical algorithms provide a powerful tool for detecting and predicting patterns as well as understanding data (Davison, 2003).

Statistical algorithms played an essential role in assessing the researchers to build algorithms for QoS and QoE correlation modelling. For instance, in a trial done by Aroussi et al. (2012), Multiple Linear Regression (MLR) was used to draw the relation between the QoS network parameters and the perceived QoE for Video on Demand (VoD) services. The researchers in this paper have proposed QoE-QoS exponential correlation algorithm that is based on MLR. Such algorithm achieved 0.8389 for the R^2 performance measure.

In a paper done by Zhang & Wang (2018), the mapping between the QoS and QoE was studied. The authors developed a statistical algorithm to estimate the MOS of the Fifth Generation of Mobile Networks (5G) mobile traffic offloading to WiFi access points. The QoE parameters were based on human aspects (age, role, gender, etc.), context aspects (environment and energy consumption), and technical aspects (QoS parameters).

With the evolvement of telecommunications networks, the demand for the multimedia content has significantly increased. Paper was written by Zhu et al. (2020) in which they have presented a statistical algorithm. Such algorithm models the QoS parameters and the MOS by considering

video bit rate and jitter in given MOS ranges. The researchers claimed that their algorithm can provide the relevant stakeholders with advanced suggestions for configuring QoS parameters at a specific user's QoE level.

2.3.2. Machine Learning Models

ML models, on the other hand, are a type of algorithms that can learn from data and make predictions or decisions without being explicitly programmed. They are often used for tasks such as classification and regression and are more complex than statistical models but can be less interpretable than statistical models. ML models are broadly categorized into supervised, unsupervised, and reinforcement learning models. ML models are a subset of artificial intelligence and have been used to solve a wide range of real-life problems, such as image recognition, natural language processing, and anomaly detection. Examples of ML models include Decision Tree (DT), Random Forest (RF), SVM, Naïve Bayes (NB), and K-Nearest Neighbour (KNN). Supervised learning models are trained on labelled data and are used for tasks such as classification and regression. Unsupervised learning models, such as clustering, are used to find patterns in unlabelled data. Reinforcement learning models are trained by receiving feedback in the form of rewards or punishments and are used in decision-making and control systems (Burkov, 2019).

One of the specializations that researchers worked on is estimating the QoE for Voice over Internet Protocol (VoIP) services. Zhang and Wang (2018) have focused their research on the spatial and temporal features of the VoIP networks deteriorations. Regression ML algorithms were implemented to estimate the QoE using the aforementioned features. The authors experimented 5 algorithms. They are KNN, Regression Tree, Artificial Neural Networks (ANN), Bagging and SVM. SVM has mostly shown better scores (lowest Mean Absolute Error (MAE)) that equals to

(0.0417) for the different testing scenarios. Synthetic datasets were generated for such study. To do so, NIST Net network emulator and OpenPhone VoIP application were utilized.

The investigation conducted by Vasilev et al. (2018) resulted in mapping the QoS parameters to three main elements (average video bitrate, average video bitrate variation and re-buffering ration) that influence the QoE the most for video-based telecom services. For the mapping the authors used RF and Logistic Regression (LR) based Bayesian Network (BN) algorithms. The later outperformed the RF with an accuracy that reached 93.68%. For the experiment, the authors used open- source movies, Big Buck Bunny, Swiss Account and Red Bull Play Street.

Furthermore, Casas et al. (2017) performed an experiment on real mobile network data collected from field experiments to predict the QoE for popular applications (e.g., YouTube and Facebook). The researchers of this paper have compared the performance of five ML models, namely, Multi-layer Perceptron (MLP), SVM, DT, NB and RF. RF achieved the highest accuracy, 98%.

For multiple telecom services (web browsing, video streaming and VoIP), Aroussi and Mellouk (2016) have conducted research to predict the perceived QoE in which multiple QoS parameters were taken into consideration (e.g., delay, jitter, packet loss, video types, users' profile, and terminal characteristics). These parameters were correlated to the QoE class that they correspond to. This is done via the training of seven ML algorithms, namely, SVM, NB, ANN, KNN, DT and RF. DT algorithm gave the best performance. It was able to correctly classify the QoE 89.64% of the time. Such study was conducted using a self-collected dataset.

2.3.3. Deep Learning Models

DL algorithms are a type of ML algorithms that use ANN with multiple layers, referred to as deep ANN. Such learning models are inspired by the structure and the function of the human neurons. The layered structure of DL models allows them to learn the hierarchy of features, which enables them to understand complex patterns in the data. There are many different types of DL models, such as feedforward ANN, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). Feedforward ANN, also known as MLP, are used for a wide range of tasks such as image recognition, speech recognition, and natural language processing. CNN are specifically designed for image recognition tasks, while RNN are used for tasks that involve sequential data such as speech recognition and natural language processing (Zhang et al., 2022a).

Recently, more studies concerned with QoE prediction are conducted using DL algorithms. In a paper prepared by Zhang et al. (2022b), video streaming QoE prediction method was created that is based on CNN and Gated Recurrent Unit (GRU), variant of Long Short-Term Memory (LSTM). The idea of this model is unique in the sense that it only considers four QoS factors (short time subjective quality, playback indicator, current bitrate and time elapsed since last rebuffering) and it discards the rest. The proposed method was tested using LFOVIA database. It produced RMSE of 7.93%.

A relatively close work to this research is a study done by (Song et al., 2020). In this study, the researchers predicted the QoE of the users visiting six web pages, namely, taobao, sina, bilibili, amazon, qq and jd. LSTM algorithm (which is a type of RNN) was the predictor of choice. Ten features were extracted from Transmission Control Protocol (TCP) packets and a traffic metric called Cumulative Traffic Volume (CTV). LSTM was fed by those features. Such features are pertaining to web page

visit time and traffic volume. The QoE prediction results of LSTM reached 0.9 for the R^2 performance measure.

Moreover, a system was created by Frnda et al. (2019) to review the characteristics that were retrieved from the Internet Protocol Television (IPTV) transmission tools. Accordingly, the system will interpret the collected data to draw a subjective conclusion about the quality of the transmitted content, i.e., mimicking human subjective evaluation of the quality. The data were related to video QoS parameters (packet loss, resolution, bitrate, codec type, and Structural Similarity Index Measure (SSIM)). To prepare the training dataset for the DL based system, 60 volunteers were asked to rate the self-generated videos according with MOS scheme. The DL algorithm that was used for the evaluation prediction was the BPNN. It achieved RMSE value of 0.19.

As an example of DL use case in indirectly improving the QoS, a trial was done by Zafar et al. (2019) in which LSTM was used to predict the location of city buses. The prediction helps in minimizing the interference between small cells mounted of these buses, hence TSP can provide better QoS. Such small cells are helping in providing better coverage for the mobile IoT. The algorithm has correctly predicted the buses' locations 98.6% of the time.

A DL based approach (ANN) is developed by Tao et al. (2019) to directly evaluate QoE perceived by end users while videos being transmitted to their mobile gadgets. For this approach, a crowdsourcing application was developed to collect 89 QoS parameters along with the users' ratings for the videos they have watched. Feature selection technique was applied to choose the most relevant features. Selected features are related to video coding format, the video bit rate, buffering delay and so on. ANN algorithm scored the lowest RMSE value (0.8686) among the

tested algorithms (SVM, DT and Adaptive Neuro Fuzzy Inference System (ANFIS)).

In the experiment described in Begluk et al. (2018) paper, the authors presented DL based software that predicts the QoE of streamed videos over the Fourth Generation of Mobile Networks (4G) network in relation to the MOS scheme. The ANN based algorithm was trained by an open-source dataset from H2O platform. Such dataset was developed based on survey. It contained network performance parameters (delay, jitter, loss, error rates and distributions, throughput, etc.) as well as the equivalent MOS score for each record. From the obtained result ($t - score = 0.104$), it can be concluded that there no difference between the predicted MOS and the MOS obtained from the dataset. Hence, the prediction model in question can predict MOS with relatively high accuracy.

A paper conducted by Gao et al. (2017) was written about predicting the QoE of the content streamed on IPTVs. This paper proposed a prediction model that is based on combined algorithms, namely, BPNN and AdaBoost. The authors went through extensive process of data preparation before they were able to feed the algorithms with input. Furthermore, the authors came with a new feature as an output (predicted QoE) that they called "viewing ratio". Which is how long the viewer kept watching a show without switching it or turn-off the service. The viewing ration was derived from network parameters. The new feature was used in conjunction with conventional network technical parameters to train the combined algorithms. The training dataset was provided by Zhongxing Telecommunication Equipment Corporation. The proposed combined algorithms outperformed other tested algorithms (SVM and BPNN). It scored the lowest MSE, which was 0.21.

To sum up, statistical algorithms are simple and interpretable, ML algorithms are more complex and perform well on complex problems, and

DL algorithms are highly complex and able to handle large amounts of data but can be difficult to interpret. Each type of algorithm has its own strengths and weaknesses as demonstrated in Table 2.3 below. The choice of which algorithm to use depends on the specificity of problem and the available data. This table helped in guiding the researcher to narrow down the algorithms that could be used.

Table (2.3): Strengths and weaknesses of the algorithms

Algorithm Type	Strengths	Weaknesses
Statistical	• simple and interpretable • based on well-established mathematical and statistical theories • performs well in small to medium-sized datasets	• may not perform well when the problem is highly complex and/or the dataset is large • rely on certain assumptions about the data, such as normality or independence
ML & DL	• can handle complex relationships between variables • able to deal with non-linear patterns in data • can sometimes automatically select which features are most important for making predictions • scale well to large datasets • able to achieve state-of-the-art performance	• can be difficult to interpret • require large amounts of data to train effectively • may not generalize well • computationally expensive, requiring specialized hardware and a lot of time

Unlike statistical algorithms, DL have proven their effectiveness in predicting QoE with relatively good level of performance. In consideration of the aforementioned, only ML and DL algorithms were used in this work to analyse the dataset of this research. This is because the nature of the dataset dictates the usage of the ML and DL algorithms. The dataset is considered to be large, and the attributes are several. Moreover, ML algorithms allow for analysing the non-linear patterns and the DL helps in

selecting the most relevant features. Nevertheless, all the above discussed algorithms were trained by batch learning technique which has some limitations.

In batch learning, the algorithm is trained on all the available data at once due to its inability of learning gradually. This approach can be less flexible when the data distribution changes over time. Also, because the algorithm is trained on all available data at once, it requires more memory, and it consumes more time to train the algorithm. Furthermore, batch learning is not easily scalable, as adding new data to the model requires retraining on the entire dataset. This makes it less suitable for applications where data is constantly being generated and needs to be added to the model in real-time. Batch learning is generally used when the data is available all at once (Kumar, 2022). However, data is not always fully available at once, which is the case for real-time applications. To overcome such shortcomings, incremental learning technique is proposed in this research.

2.4. Incremental Learning

As per the study conducted by Kefi-Fatteh et al. (2019), in incremental learning, the algorithm is trained on small amounts of data at a time and its parameters are updated incrementally as new data becomes available. Nonetheless, the aforementioned study adds that unlike batch learning, incremental learning allows the algorithm to continuously adapt to new data and improve over time and that it is more effective when processing real-time or large amount of data. Incremental learning can be more efficient than batch learning since it only needs to update the algorithm with the new data rather than retraining the entire algorithm from scratch. This can save time and computing resources. Incremental learning

is generally used in applications where data is coming in a steady stream, and the model needs to adapt quickly to new data.

Recently, only handful of papers are published about experiments that the incremental learning approach was part of. This is specifically true for publications about QoE prediction. In a paper published couple of years ago by Elwerghemmi et al. (2019), a novel QoE predictor was introduced, namely Stacked Incremental Support Vector Machine (SISVM). It is a combination between stacking model and Incremental Support Vector Machine (ISVM). In this research, ISVM is utilized to gradually assess the different aspects of the issue and then produce the output based on a probabilistic approach. The learning datasets are video content extracted from Poqemon-QoE-Database. Such datasets consisted of 22 parameters related to network performance parameters, gadgets hardware information as well as content parameters like resolution and audio-loss in addition to the subjective QoE rating that is based on MOS. The performance of SISVM was compared with the performance of SVM, ANN, RF and Adaboost Decision Tree (ADT). SISVM has outperformed them with an accuracy of 81.84%.

Another research that incorporated the incremental learning in its experiment was conducted by Ben Youssef et al. (2018). Multiclass incremental SVM was developed. The algorithm was designed to classify users' perception of video streaming services. 45 individuals participated to evaluate and classify the quality of 24 videos. The classes were based on MOS scheme. The algorithm was trained on features pertaining to network parameters and content parameters, including but not limited to video size, frame rate, frame lost, buffer time, resolution, etc. The algorithm was able to achieve an overall accuracy of 89.3%, with initialising the incremental process that is based on only 10% of the dataset. The ISVM outpaced the DT, RF, RNN and SVM algorithms.

Table 2.4 below summarizes the studies related to QoE prediction using ML and DL algorithms.

Table (2.4): Overview of studies on QoE prediction using ML and DL algorithms

Citation	Studied Dataset	Area of Study	Best ML Algorithm	Performance Measure	Performance Value	Learning Approach
(Zhang et al., 2022)	LFOVIA	Video Streaming	CNN+ LSTM	RMSE	7.93%	Batch
(Hu et al., 2020)	Self-generated	VoIP	SVM	MAE	0.0417	
(Song et al., 2020)	Self-generated	Web Browsing	LSTM	R^2	0.9	
(Frnda et al., 2019)	Self-generated	IPTV	BPNN	RMSE	0.19	
(Tao et al., 2019)	Self-collected	Video Streaming	DNN	RMSE	0.8686	
(Zafar et al., 2019)	Self-generated	Coverage Enhancement	LSTM	Accuracy	98.60%	
(Begluk et al., 2018)	Open-source H2O patform	Video Streaming	ANN	t-score	0.104	
(Shan et al., 2018)	Beijing Institute of Technology	Speech Quality	BPNN	RMSE	0.4612	
(Vasilev et al., 2018)	Open-source movies	Video Streaming	RF	Accuracy	93.68%	
(Casas et al., 2017)	Self-collected	Web Browsing	RF	Accuracy	98%	
(Gao et al., 2017)	Zhongxing Telecommuni ca-tion Equipment Corporation	IPTV	BPNN+ AdaBoost	MSE	0.21	

Table (2.4): Overview of studies on QoE prediction using ML and DL algorithms (cont.)

Citation	Studied Dataset	Area of Study	Best ML Algorithm	Performance Measure	Performance Value	Learning Approach
(Aroussi & Mellouk, 2016)	Self-collected	Web Browsing, Video Streaming and VoIP	DT	Accuracy	89.64%	Batch
(Elwerghemmi et al., 2019)	Poqemon-QoE-Dataset	Video Streaming	SISVM	Accuracy	81.84%	Incremental
(Ben Youssef et al., 2018)	Open-source movies	Video Streaming	ISVM	Accuracy	89.30%	

Throughout the conducted research about the incremental learning, it has been noticed that SVM algorithms that have been trained incrementally have performed relatively better than the other algorithms, whether the others are trained in batch or incrementally. For instance, in research by Soula et al. (2020), a novel incremental Kernel Nonparametric SVM algorithm was introduced that had the capability of handling the data spread issues as well as learning gradually. The algorithm in question was tested using both, real world datasets and artificial dataset. It has, predominantly, outperformed other seven batch-based and incremental-based learning algorithms.

One more research that utilized the incremental learning capabilities was conducted by (Kefi-Fatteh et al., 2019). In this research one-class classifier was tested that underlines the training dataset's low variance orientation, which leads to greater accuracy. It was called incremental Covariance-guided One-Class SVM. Like the previous paper, the one-class classifier was tested using both, real world datasets (medical and handwritten digits) and artificial dataset. Overall, it has performed better than the other batch-based and incremental-based learning methods.

In earlier paper by Losing et al. (2016), the incremental learning subject was addressed. Authors of this paper have compared 7 incremental

classifiers, all of which were trained and evaluated using 10 datasets of various complexities. Again, the ISVM classifier has shown better performance than the other 6 classifiers. This was the case for 8 of 10 datasets.

In light of the witnessed superiority of the SVM's performance if trained incrementally, it has been decided that the dataset studied in this research will be analysed through an incremental SVM model.

2.5. Summary

A novel classifier was experimented in the research conducted by (Salhi et al., 2022). They have incorporated the discrimination approach via Kernelized Non-Parametric Discriminant Analysis (KNDA) algorithm to find the optimal combination of features that best separates the different classes of data. The SVM and KNDA were trained using the incremental learning approach to accommodate the dynamic nature of the examined dataset. This assisted with overcoming the memory and time constraints.

Considering the advantages mentioned above, this research followed the novel compound model IDSVM that the paper Salhi et al. (2022) is evolving around. From the covered literature and to the best of our knowledge, it can be claimed that this novel classifier (IDSVM) has not been experimented in the QoE prediction domain.

Chapter Three

Methods and Procedures

3.0. Methods

3.1. Procedures

Chapter Three

Methods and Procedures

3.1. Methods

QoE is a crucial factor for evaluating the performance of telecom services. In recent years, ML algorithms have been increasingly employed for QoE prediction. Among these algorithms, SVM have been widely adopted due to their high classification accuracy. However, SVMs may be computationally expensive and memory intensive for large datasets. To address these issues, incremental learning has been implemented, which updates the algorithms incrementally as new data points become available. Additionally, KNDA has shown to complement the classical SVM shortcoming of building the hyperplane based on the local properties of the data, as global properties of data will be taken into consideration as well. In this context, this research proposes the combination of SVM and KNDA along with the incremental learning approach for QoE classification. Fig. 3.1 demonstrates the combined models that is used in this research.

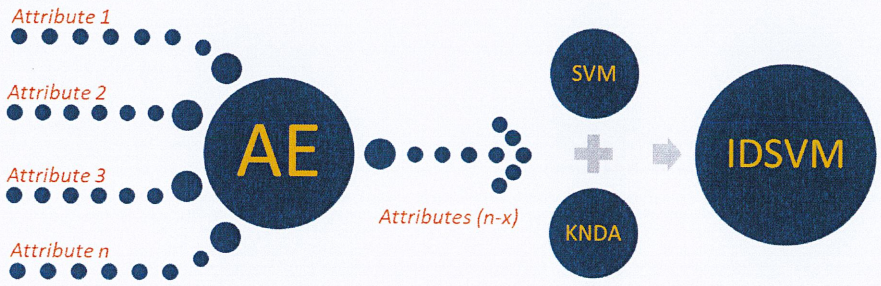


Fig. 3.1: Compound novel model

3.1.1. IDSVM

The IDSVM compound model is, as mentioned previously, based on the paper (Salhi et al., 2022). In this research, the IDSVM has been used in conjunction with the AE for dimensionality reduction purposes. Although the dataset studied in this paper is related to the medical field, the advantage of the incremental learning along with the advantages of the involved algorithms have triggered the idea of experimenting the IDSVM on a different dataset that is related to predicting the QoE for telecom services.

IDSVM is based on the basic forms of the SVM and discriminant analysis. It is considered as an enhancement to these algorithms. In particular, the enhancement lies within the model's ability to combine both the global and local variants offered by the maximum margin-based and discriminant-based approaches. In addition, it uses the input space rather than the feature space to generate the variance-covariance matrix. By doing this, it gets the near-global data variations in the least time possible. Furthermore, IDSVM makes use of active learning as new data points are made available to the model. In addition, it effectively handles data that may be separated into linear and non-linear forms. IDSVM is also considered time efficient classifier that requires relatively small memory.

3.1.1.1. DSVM

The benefits of the discriminant-based classifiers and the SVM classifier are combined to drive the Discriminant based Support Vector Machine (DSVM) classifier. The between-class and within-class variance-covariance matrices, computed in the input space, are blended by adding them to the SVM optimization problem. By taking into account the overall information offered by discriminant-based classifier (KNDA) and computing the variance-covariance matrices in the input space, it is feasible to get around the time and memory storage constraints. Fig. 3.2

demonstrates what is meant by between-class and within-class in the discriminant-based classifier.

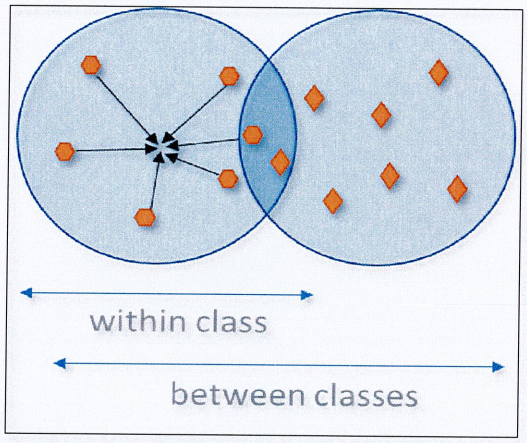


Fig. 3.2: Between-class and within-class graphical illustration

The performance will then be enhanced by fusing global features with local ones offered by SVM (i.e., the consideration is going beyond the support vectors) to generate more effective and reliable results than baseline SVM, as shown in Fig. 3.3. The proposed method considers the near-global variations of the class distribution. We employ the KNDA equations to calculate the between-class and within-class variance-covariance matrices, but we do so in the input space rather than the feature space because in the latter, the dimension is equal to the number of data points, making the calculation more difficult and time-consuming.

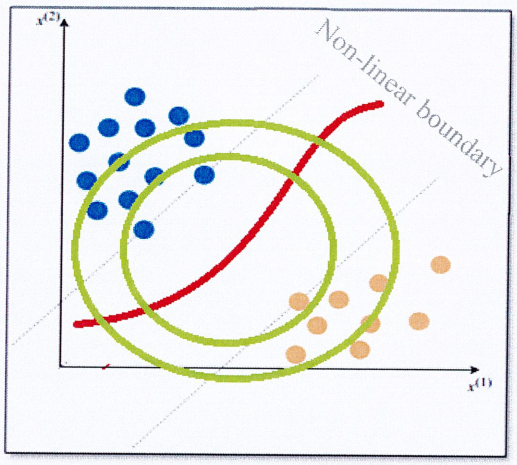


Fig. 3.3: Non-linear boundary and near-global support vectors in SVM. Adapted from (Burkov, 2019)

Equations demonstrated below that are pertaining to the IDSVM model are from the paper (Salhi et al., 2022). In order to draw up the between-class variance-covariance matrix, we calculate the K nearest neighbour average vectors instead of simple average vectors in the input space. The model is designed to stop when we find the K that produces the best value (optimal K). The between-class variance-covariance matrix is calculated via the following Equation (3.1):

$$VCM_b = \frac{1}{M_1 + M_2}\left(\sum_{i=1}^{M_1} \beta_i Q1(i)^S Q1(i) + \sum_{i=1}^{M_2} \beta_i Q2(i)^S Q2(i)\right) \quad (3.1)$$

Where M_x are the samples belonging to two different classes, β represents the weighting functions used to void the impacts attributable to samples that are far from the decision boundary. As for $Q1$ and $Q2$, they are given by the following Equations (3.2 and 3.3):

$$Q1(i) = a_i - L_2^k(a_i) \quad (3.2)$$

$$Q2(i) = a_i - L_1^k(a_i) \tag{3.3}$$

Where a_i refers to a data point, K is the nearest neighbour/s to that data point and L_x^k is the class that a data point belongs to.

The identification of near-global points is done via Equation (3.4) below:

$$\beta = \frac{min\left(d\left(a_i, L_1^k(a_i)\right)^{\omega}, d\left(a_i, L_2^k(a_i)\right)^{\omega}\right)}{d\left(a_i, L_1^k(a_i)\right)^{\omega} + d\left(a_i, L_2^k(a_i)\right)^{\omega}} \tag{3.4}$$

As the sample becomes distant from the decision boundary, values are becoming closer to 0. To manage how quick this occurs, we employ the parameter ω. The within-class variance-covariance matrix is also computed in the input space in accordance with Equation (3.5) below:

$$VCM_w = A_1(J - 1_{M_1})A_1^S + A_2(J - 1_{M_2})A_2^S \tag{3.5}$$

where J is the identity matrix, 1_{M_1} is the matrix with all entries $\frac{1}{M_1}$ and 1_{M_2} is the matrix with all entries $\frac{1}{M_2}$.

The matrix that incorporates the within-class variance-covariance matrix along with the between-class variance-covariance matrix is mathematically represented by Equation (3.6):

$$\varphi = \eta VCM_w\,(VCM_b + \alpha J)^{-1} VCM_w + J \tag{3.6}$$

Where η is a parameter that is used to restrict the weight vector orientation and α is a coefficient deployed to enable the inversion of the matrix.

The SVM solution is based on a minimization problem. Therefore, after plugging in the matrix (φ) into the SVM optimization problem, the proposed objective Criterion (3.7) to be minimized is given by:

$$\min_{v\neq 0, v_o} \left\{ \frac{1}{2} v^s \varphi v + C \sum_{i=1}^{M} max\,(0, 1 - b_i\,(v^s a_i + v_0) \right\} \quad (3.7)$$

Where v is the weight and C is the margin size in the SVM algorithm, while b_i is the associated class of the data point. During the training, the objective criterion will get optimized with respect to the hyperparameters.

The variable change is used to calculate the matrix in the input space. In essence, it is a transformation that, rather than calculating in feature space, computes DSVM in the input space and then the necessary kernelization will take place. $K_{DSVM} = A^s \varphi^{-1} A$ is the new kernel matrix created for the DSVM model and $A = [a_1, .., a_M]$ is the initial training data matrix of the samples. At this stage, the classifier can only handle linearly separable data points. To enable it to handle non-linearly separable data points, Singular Value Decomposition (SVD) is applied on φ and break it down as per Equation (3.8):

$$\varphi = Z^s \Sigma Z \quad (3.8)$$

Where Z indicates the orthonormal matrix with all the characteristic vectors of φ, and Σ indicates the diagonal matrix that includes all the related characteristic roots. Next, the transform: $R = \Sigma^{-\frac{1}{2}} Z$ is implemented

to the samples a_i to compute the new data point $a_i' = Ra_i$. The transform will be executed to the overall data matrix. This is done via Equation (3.9):

$$A'^S A' = A^S R^S R A = A^S \varphi^{-1} A \tag{3.9}$$

Should baseline SVM be applied with data points a_i' and $A'^S A'$ as kernel matrix, this equals to getting the results from DSVM. Moreover, calculating baseline SVM with non-linear kernels to the transformed data points a_i' results in, applying non-linear transforms on the scalar products $a_i'^S a_j'$ that appear in Equation (3.9) pertaining to kernel matrix.

The Radial Basis Function (RBF) kernel is mathematically expressed as follow (Equation 3.10):

$$K_{DSVM-RBF}(a_i, a_j) = e^{-\frac{(a_i - a_j)^S \varphi^{-1} (a_i - a_j)}{2\sigma^2}} \tag{3.10}$$

Where σ is the variance of RBF. It changes the shape of the separation.

3.1.1.2. Incremental Learning

The aim is to convert the DSVM into an incremental algorithm that improves upon an existing optimum solution by adding one data sample a_i at a time. Repeatedly constructing the solution will increase its capacity to handle complicated and big datasets - which is the case with studied dataset in this research - while maintaining the advantage of the suggested novel classifier.

Prior to including a new point a_i to the training set, the Karush–Kuhn–Tucker optimality conditions (KKT) are met for all existing training examples. The new sample's weight is originally set to zero when it is introduced, making it a non-support vector. The weights and bias will be updated if the new point is a support vector until the KTT requirements are once more met. To implement that, we will divide A into sets based on the KTT Conditions:

1. The first set D holds the unbounded support vectors (located on the margin). The points belonging to this set have weight values between 0 and C.
2. The second set R indicates bounded support vectors that have weights equal to C.
3. The last set M holds the non-support vector points that have weight values equal to 0.
4. $O = R \cup M$

The different datasets are graphically demonstrated in Fig. 3.4 below:

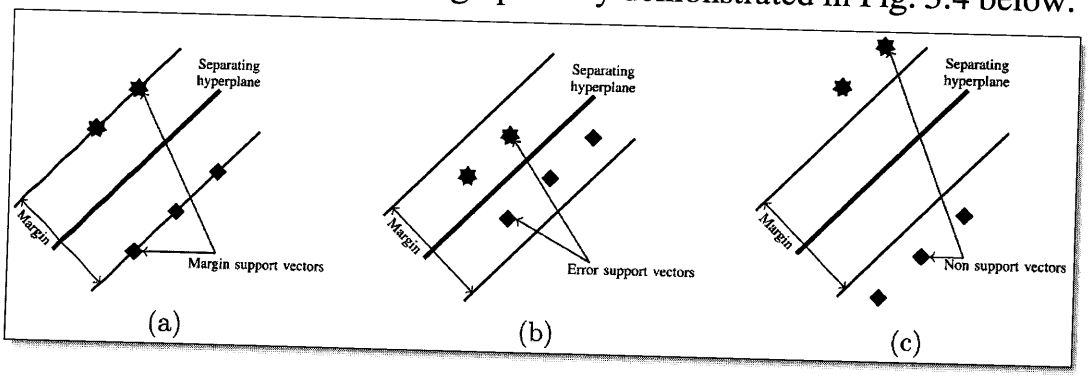


Fig. 3.4: Depiction of the subsets (a) ***D***, (b) ***R***, (c) ***M*** as given by the KTT. Adapted from (Soula et al., 2020)

The updating criteria is built on four possible scenarios as follow:

1. Number of weights μ_i in D get as far as the lower edge 0 or the upper edge C. Let θ be a minimal number, the largest gain $\Delta\mu_c^S$

that can be possible before the relocation of some example from D to O is given by the criterion (3.11):

$$\Delta\mu_c^D = \begin{cases} \frac{C-\mu_i}{\beta_i}, & if\ \beta_i > \theta \\ \frac{\mu_i}{\beta_i}, & if\ \beta_i > \theta \end{cases} \tag{3.11}$$

2. Some slope f_i in O reach 0. The largest gain $\Delta\mu_c^D$ before some examples relocate from O to D is calculated as per Equation (3.12):

$$\Delta\mu_c^O = \frac{-f_i}{\rho_i} \tag{3.12}$$

3. The slope of a_c becomes 0. Equation (3.13) below calculates the maximum possible gain $\Delta\mu_c^f$:

$$\Delta\mu_c^f = \frac{-f_c}{\rho_c} \tag{3.13}$$

4. The weight of a_c becomes C. The maximum step $\Delta\mu_c^\mu$ is given by Equation (3.14):

$$\Delta\mu_c^\mu = C - \mu_c \tag{3.14}$$

Lastly, the least of the numbers for the four scenarios are used to get the maximum possible gain of μ_c following Equation (3.15):

$$\Delta\mu_c^{max} = min(\Delta\mu_c^D, \Delta\mu_c^O, \Delta\mu_c^f, \Delta\mu_c^\mu) \tag{3.15}$$

Once the $\Delta\mu_c^{max}$ is determined, the inversed matrix is re-calculated.

3.2. Procedures

Under the procedures section, a thorough explanation is provided about the phases followed to prepare the dataset for classification. A high-level flowchart (Fig. 3.5) is drawn to graphically illustrate the undertaken steps.

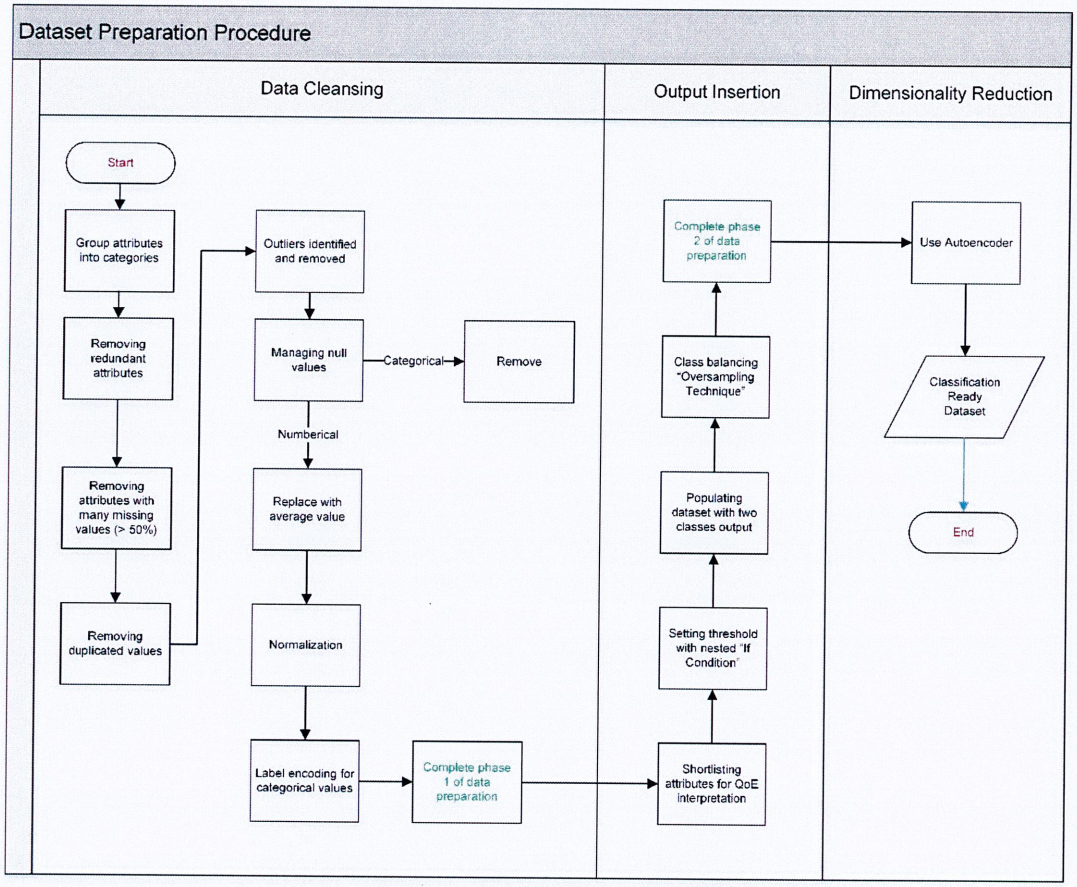


Fig. 3.5: Data preparation procedure flowchart

3.2.1. Data Cleansing

Data cleansing is an imperative phase in data analysis domain. It is nearly impossible to find data that does not require any sort of cleansing and preparation. The cleansing is done to secure a better performance of the ML algorithm as stated in the paper published by (Li et al., 2019). Only 23 attributes were chosen from the 112 attributes. The elimination process started by grouping the attributes into categories. Within each category,

redundant attributes were eliminated as well as attributes with many missing values (>50%).

The next phase was to clean the 23 attributes. The cleansing started by removing duplicated values. Then outliers were manually identified and removed based on subject matter expert consultation. The removed outliers were belonging to WiFi Received Signal Strength Indicator (RSSI), full webpage load time and average jitter attributes. As for the null values, it had to be treated based on the attribute type. For categorical attributes, records pertaining to those null values were removed. However, for the numerical attributes, null values were replaced with mean values. Upon completing the cleansing, the dataset was split into three datasets, each related to a social media platform and had around half a million record. Upon performing the cleansing, datasets had 23 attributes and approximately half million records, per social media platform. It started with 112 attributes and 2 million records per social media platform.

To ensure numerical attribute values range within a similar scale, normalization was applied following the Equation (3.16) below from (Zack, 2020).

$$Normalized\ Value = \frac{x\ value - \min value}{\max value - \min value} \qquad (3.16)$$

AE cannot directly work with categorical values. They must be first converted to numerical values (Delong & Kozak, 2022). To represent the categorical values into numerical values, encoding technique was adopted. Specifically, label encoding was applied from the scikit-learn python library. Also, as a part of the data preparation, the correlation among the attributes was examined to ensure an adequate training of the ML models.

It was found that the correlation between the attributes is minimal (i.e., very distant from 1). Hence, the 23 attributes were maintained.

3.2.2. Output Insertion

The data was collected by the Telecommunications Regulatory Authority to present main QoS parameters to the consumers to empower them to make informative decisions about their TSP. Naturally, the data lacked the QoE subjective indicator (the output). Therefore, the QoE had to be calculated based on certain thresholds. To determine such thresholds, extensive research took place to decide whether a particular record represents an acceptable QoE or not.

Among the 23 attributes, only 10 contributed to the determination the QoE. It was not possible to set thresholds for the remaining attributes due to their categorical nature. Moreover, after considering the research paper done by Saverimoutou et al. (2019), the 10 attributes were found to be common parameters in gauging the experience of web browsing. Table 3.1 below demonstrates the adopted thresholds and the source thereof.

Table (3.1): Threshold per attribute

Name of Attribute	Acceptable Range	Unit of Measure	Threshold Source
WiFi RSSI	>= -70	dBm	(Screenbeam, n.d.)
DNS Lookup	<= 120	millisecond	(Tutela, 2021)
TTFB	<=1200	millisecond	(Littledata, 2022)
Webpage Full Load Time	<=6800	millisecond	(Sematext, n.d.)
Average Jitter	<=20	millisecond	(Sematext, n.d.)
Average Ping	<=100	millisecond	(Sematext, n.d.)
Average Download Speed	>= 1.5	Mbps	(Sematext, n.d.)
Average Upload Speed	>= 0.5	Mbps	(Sematext, n.d.)
4G RSSI	>= -100	dBm	(Teltonika, 2022)
3G RSSI	>= -85	dBm	(Teltonika, 2022)

Nested "IF" function was used to produce the output (QoE). The function's conditions were based on the acceptable range of each of the 10 attributes. If one or more acceptable range is not met, the QoE is then deemed inacceptable. Then the dataset was populated with the output. It was found out that the labels were imbalanced. The records that yielded inacceptable QoE were 70% of the overall records. Such imbalance necessitated the usage of balancing technique, more precisely, over sampling technique.

3.2.2.1. Oversampling Technique

Adaptive Synthetic Sampling (ADASYN) was the technique of choice. It was indicated in a paper written by He et al. (2008) that ADASYN is a data oversampling technique designed to address the class imbalance problem in ML, where the number of samples in one class is smaller than the other. It generates synthetic data points for the minority class by computing the density distribution of the minority class and generating new samples in the regions with lower density. Moreover, He et al. (2008) authors' compared ADASYN to other oversampling techniques, such as random oversampling, Synthetic Minority Over-sampling Technique (SMOTE) and their variants, and indicated that ADASYN has some advantages as follow:

1. It adjusts the number of synthetic samples to generate for each minority class instance based on the degree of imbalance in the dataset. This means that it can create more synthetic samples for the minority class instances that are harder to learn, which can improve the performance of the classifier.
2. Unlike SMOTE, which create synthetic samples by interpolating between existing minority class samples, ADASYN generates

synthetic samples in the regions with lower density of the minority class. This helps to avoid overfitting, as the synthetic samples are more diverse and less likely to be too similar to the original minority class samples.

Upon applying ADASYN, the classes (acceptable and inacceptable QoE) were fully balanced.

3.2.2.2. ADASYN Mathematical Representation

As explained in a paper written by Huang (2015), the ADASYN works in accordance with the following steps and equations. Let X be the feature space and Y be the target variable with $Y = 1$ for the minority class and $Y = 0$ for the majority class. Let D be the dataset with N samples and d_i be the i_{th} sample in D with its corresponding label y_i.

1. Determine the ratio of the difference between the number of samples in the majority and minority classes according to Equation (3.17):

$$\gamma = \frac{(N_{majority} - N_{minority})}{N_{minority}} \tag{3.17}$$

2. For each minority sample d_i, find its KNN in X as per Equation (3.18):

$$NN_i = d_j | d_j \in D, y_j = 1, j \neq i \tag{3.18}$$

3. Equation (3.19) computes the density distribution of the minority class samples based on their distances to their KNN:

$$di = \sum_{d_j \in NN_i} d(X_i, X_j)/k \tag{3.19}$$

where $d(X_i, X_j)$ is the distance between samples X_i and X_j.

4. For each minority sample d_i, generate γ synthetic samples in the regions of the feature space following Equation (3.20) where the density of minority samples is lower than a predefined threshold:

$$S_i = s_j | j = 1, \ldots, \gamma \tag{3.20}$$

where s_j is a synthetic sample generated by interpolating between d_i and a randomly selected sample from its kNN with a weighting factor proportional to their distances.

5. Combine the original minority samples with the newly generated synthetic samples via Equation (3.21) to form the augmented minority class:

$$D' = D \cup S_1, S_2, \ldots, S_N \tag{3.21}$$

where $N = \gamma * N_{minority}$ is the total number of synthetic samples generated.

3.2.3. Dimensionality Reduction

The QoS related parameters in mobile telecom networks are enormous. Considering all the parameters in predicting the QoE is not a viable option, neither an efficient one. Only the parameters that influence the perceived QoE the most must be selected. Therefore, in this research,

AE was chosen to be the algorithm that will only maintain the highly influencing features through the dimensionality reduction technique.

In a chapter written by Bank et al. (2020), the AE was described as a type of ANN that is used for unsupervised learning, which is capable of learning patterns in data without the need for explicit labels. The goal of an AE is to learn a compressed representation or encoding of the input data in a lower-dimensional space. This compressed representation can then be used to reconstruct the original input data, which allows the AE to act as a powerful tool for data compression and denoising.

As shown in Fig. 3.6, AEs' architecture consists of two main parts: an encoder and a decoder, both of which contain one or more hidden layer. These parts are symmetrical in shape. The encoder network takes in an input data point, which is typically a vector, and applies a series of linear and nonlinear transformations to the data. The encoder will map the input to a lower-dimensional representation via the latent space (also known as bottleneck), while the decoder takes this lower-dimensional representation and maps it back to the original data space. The AE is trained by minimizing the difference between the original input data and the reconstructed data, which encourages the network to learn a compressed representation that captures the most important features of the input data.

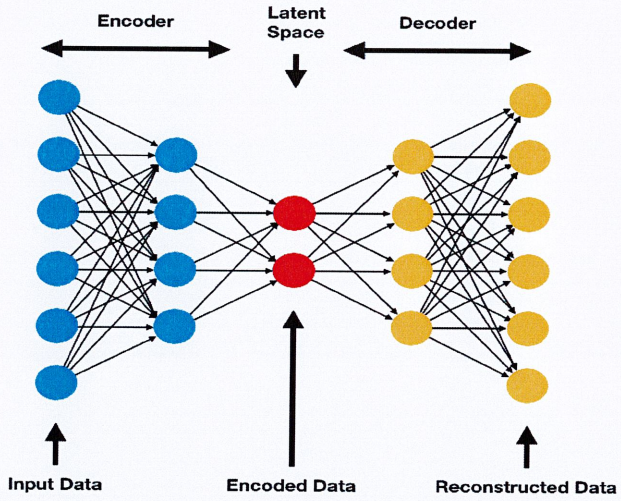


Fig. 3.6: Illustrative AE Architecture. Adapted from (Flores, 2019)

The hidden layers in the AE contain activation function. Generally, the activation functions will use mathematical operations to determine if the neuron's input to the network is significant or not when making predictions. Rectified Linear Unit (ReLU) function is a popular choice for the hidden layers of AEs. It is a simple, yet effective activation function that sets all negative inputs to 0 and leaves positive inputs unchanged. The ReLU activation function is defined as per the mathematical Equation (3.21):

$$f(x) = max(0, x) \tag{3.21}$$

where x is the input to the function and $max(0, x)$ returns the maximum of 0 and x.

AEs are very effective at dimensionality reduction added (Bank et al., 2020). It can reduce the number of input features or variables in a dataset while retaining the most important information. Once the AE is trained by passing the dataset through the encoder and decoder, dimensionality

reduction is achieved by passing the dataset again, but this time through the encoder and bottleneck parts only.

By reducing the number of input features, the AE can also help to reduce overfitting and improve the efficiency and accuracy of other machine learning algorithms that are trained on the reduced dataset (Bank et al., 2020). The diagram below (Fig. 3.7) depicts the general stages that have been followed in this research for data preparation and model building.

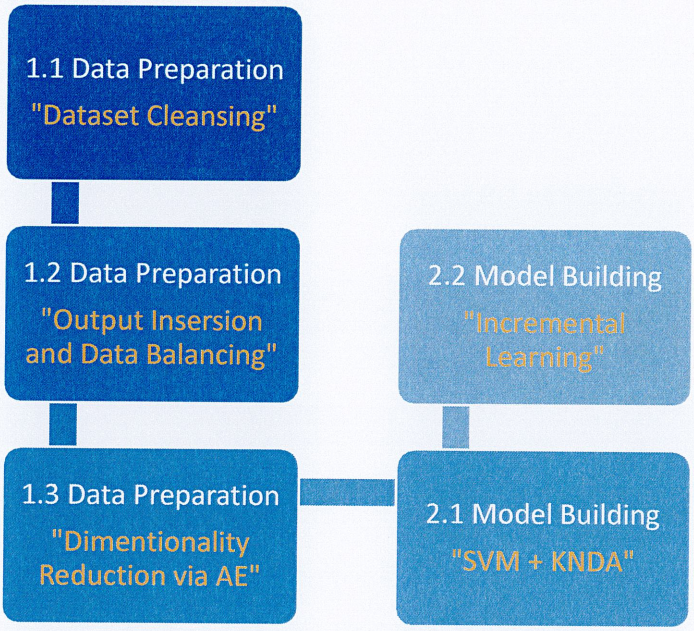


Fig. 3.7: Steps followed for data preparation and model building

Chapter Four

Experiments, Results and Discussion

4.1. Dataset Description

4.2. Experiment Settings and Protocols

4.3. Comparative Algorithms

4.4. Performance Measures

4.5. Results and Discussion

Chapter Four

Experiments, Results and Discussion

4.1. Dataset Description

In this section, we will touch upon the dataset studied in this research. Considering that this is the first time that such dataset is made available for research and experiment, it is essential to describe it with certain level of details.

The dataset is built by accumulating data collected from smart mobile gadgets via crowdsourcing. As stated in ITU's recommendation Telecommunications Standardization Sector of ITU (2020), crowdsourcing is defined as a service performed by large number of people through online means. The data is generated by running background tests through some of the applications that are already installed in the smart mobile gadgets. The dataset contains test results pertaining to browsing the homepage, the login page and the public social profile page of Facebook, Twitter, and Instagram. Considering the restrictions that Apple imposes on the background testing, the dataset contains tests from Android based smart mobile gadgets only.

The dataset contained 112 attributes with approximately 2 million records per social media platform. The overall size of the dataset was 6.22 GB. The dataset was extracted in JavaScript Object Notation (JSON) format. The data collected covered two years of measurements, 2021 and part of 2022. The attributes were both, numerical and categorical in nature. The attributes could be widely categorized into 10 categories as shown in Table 4.1 below. Each attribute provided information about certain aspect

of the conducted tests. Considering that the tests are geographically scattered and considering that number of the conducted tests (approximately 2 million records per platform), it is believed that the dataset represents reasonable sample of the population and hence generalization to the population of Bahrain can be deduced.

Table (4.1): Attribute category description

Attribute Category	Description
Location	Longitude, latitude and name of city/town
Date & Time	The calendar day at which the test was conducted along with the time in seconds, minutes and hours
Hardware	The smart gadget manufacturer and model
Software	Application name & version, SDK version, Android API and Operating system version
Connection Type	WiFi, 3G, or 4G
Radio Condition	Signal level in dBm
Network	IP address and Autonomous System Number
Service Provider	Name of the mobile service provider
Signal Timing	Full webpage load time, Average ping, Average jitter, Time to first byte and DNS lookup time
Speed	Average download speed and Average upload speed

4.2. Experiment Settings and Protocols

As stated in Chapter 2, the compound model consisted of three algorithms, AE, SVM and KNDA. To enable these algorithms to reach to the best performance possible, hyperparameters were adjusted with several attempts to tune the algorithms. Following, the hyperparameters pertaining to these algorithms will be explained and values that produced the best performance will be presented. To build the AE algorithm, Python 3.7 was

used via the cloud solution Google Colab. As for building the SVM and KNDA, MATLAB R2022a was used.

4.2.1. Autoencoder Hyperparameters

Below are the hyperparameters than have been adjusted for the AE algorithm. The adjustments were following the trial-and-error method:

- **Hidden Layers**

Are layers in ANN that are not directly connected to the input or output layers. The number of hidden layers and the number of neurons in each hidden layer are hyperparameters that can be adjusted to optimize the performance of ANN algorithm.

- **Neurons**

A neuron is a processing unit that receives input signals, performs a computation, and produces an output signal. The number of neurons in a layer is a hyperparameter that determines the complexity and capacity of an ANN.

- **Activation Function**

Is a function applied to the output of each neuron in an ANN. It introduces non-linearity into the algorithm and enables the ANN to learn complex patterns in the data. The choice of the activation function type is a hyperparameter that can affect the performance of an algorithm.

- **Learning Rate**

A hyperparameter that determines the step size at each iteration while moving toward a minimum of a loss function during the training of a ML

algorithm. A high learning rate can result in overshooting the minimum, while a low learning rate can result in a slow convergence to the minimum.

- **Weight Decay**

Weight decay is a regularization technique that adds a penalty term to the loss function to discourage large weight values. It can prevent overfitting and improve the generalization performance of ANN algorithm. The weight decay parameter is a hyperparameter that determines the strength of the penalty term.

- **Batch Size**

The number of training samples used in one forward/backward pass during the training of a machine learning model. The batch size hyperparameter can affect the convergence speed and the memory requirements of the training process.

- **Epochs**

Epochs refer to the number of times a ML algorithm will process the entire training dataset. The number of epochs is a hyperparameter that determines how long a model will be trained for.

- **Data Split**

Data split is the process of dividing a dataset into subsets for training and testing. The proportion of data allocated to each subset is a hyperparameter that can affect the algorithm's performance and generalization ability.

Table 4.2 below lists the hyperparameters values that allowed the AE to provide the best performance for the 3 datasets (Facebook, Twitter and Instagram).

Table (4.2): AE hyperparameters for top performance

Hyperparameter	Value
Architecture (Encoder)	Input:23 Neurons Hidden Layer #1: 20 Neurons Hidden Layer #2: 18 Neurons Bottleneck: 16 Neurons
Activation Function	ReLU
Learning Rate	0.0001
Weight Decay	0.0001
Batch Size	1000
Number of Epochs	1000
Data Split	Training: 90% Testing: 10%

The values pertaining to the encoder architecture in Table 4.2 are replicated in a symmetric way for the decoder part (i.e., in a reverse manner) without the bottleneck. Bottleneck of size 16 produced better results in general, hence it was fixed and number of neurons and hiddens layers were manipulated. As for the activation function, ReLU was found to be simple function that is computationally efficient to evaluate, making it a good choice for large-scale models and datasets. Moreover, ReLU is able to avoid the vanishing gradient problem, which can occur when gradients become very small during backpropagation. This problem can make it difficult for neural networks to learn and can slow down training, but ReLU's derivative is either 0 or 1, which avoids this issue. Furthermore, ReLU is a nonlinear function, which allows AEs to learn

complex, nonlinear relationships between the input and output (Bai, 2022). It was noticed that the smaller the learning rate and weight decay the better the performance of the AE. The data split was 90% for training and 10% for testing. This split is believed to be adequate due to the dataset size.

4.2.2. DSVM Hyperparameters

Below are the hyperparameters that have been adjusted for the DSVM algorithm. The adjustments were done in accordance with the 10-fold cross validation method:

- **Regularization parameter "C"**

In SVMs, the regularization parameter C controls the trade-off between maximizing the margin and minimizing the classification error on the training set. A larger value of C leads to a narrower margin and higher penalty for misclassification, whereas a smaller value of C allows more misclassifications but a wider margin.

- **Gamma Parameter "γ"**

The gamma parameter γ in the RBF kernel is a hyperparameter that controls the width of the kernel. It is a scalar value that determines the degree of influence that each training example has on the prediction of a new data point.

Specifically, the γ determines the smoothness of the decision boundary generated by the RBF kernel. A smaller γ value creates a smoother decision boundary, while a larger γ value leads to a more complex and wigglier boundary.

- **Nearest Neighbors "K"**

The hyperparameter k determines the number of nearest neighbors used for classification. A larger value of k makes the decision boundary smoother and tends to reduce the noise in the data, whereas a smaller value of k leads to a more complex decision boundary and tends to overfit the data.

- **10-fold Cross Validation**

The 10-fold cross-validation is a technique in ML for algorithm evaluation and selection. Such thing will ensure the highest performance by the algorithm. The technique involves splitting a dataset into ten equal parts or folds. The algorithm is trained on nine folds and tested on the remaining fold. This process is repeated ten times, with each fold being used as the test set once.

The steps of the 10-fold cross validation are:

1. Divide the dataset into ten equal parts or folds.
2. For each fold, train the algorithm on the other nine folds.
3. Evaluate the algorithm's performance on the current fold.
4. Record the performance measure (e.g., accuracy, precision, recall, etc.) for the current fold.
5. Repeat steps 2 to 4 for all ten folds.
6. Calculate the average performance measure across all ten folds to estimate the algorithm's generalization performance.

The 10-fold cross-validation can help to reduce the variance of the performance estimate compared to a single train-test split. It also allows for a more reliable estimate of the algorithm's performance on new, unseen data. Additionally, it can be used for hyperparameter tuning, where

different combinations of hyperparameters can be evaluated on the training set and tested on the validation set within each fold.

Table 4.3 below lists the hyperparameters values that enabled the IDSVM algorithm to provide the best performance for the full datasets. For the γ parameter, the value for all the datasets was the same. However, for the other two hyperparameters, the values were generally different per dataset. The 10-fold cross validation technique was applied across all datasets for data split and for choosing the optimal hyperparameters.

Table (4.3): IDSVM hyperparameters for top performance

Dataset	Regularization Parameter "C"	Gamma Parameter "γ"	Nearest Neighbors "K"
Facebook	2048	32	5
Twitter	512	32	5
Instagram	1000	32	7

Table 4.4 below lists the hyperparameters values that were chosen for the compared algorithms. The 10-fold cross validation technique was applied across all datasets and algorithms for data split and for choosing the optimal hyperparameters. The observed variations in the hyperparameters could potentially be attributed to the fact that social media platforms can have distinct characteristics in terms of the type of data. Moreover, user behaviour and content on different social media platforms can vary significantly. Furthermore, social media platforms often provide unique features and functionalities that can affect ML algorithm performance. Also, the choice of hyperparameters can also be influenced

by the technical implementation and infrastructure considerations of each social media platform.

Table (4.4): Hyperparameters for compared algorithms

Algorithms per Dataset	Regularization Parameter "C"	Gamma Parameter "γ"	Nearest Neighbours K
Facebook			
IDSVM	0.1	0.0025	3
IKNSVM	0.1	0.1	5
IKSVM	0.1	0.001	
KSVM	0.1	1	
Twitter			
IDSVM	0.001	0.0025	15
IKNSVM	0.01	25	17
IKSVM	0.01	1000	
KSVM	2048	0.0025	
Instagram			
IDSVM	0.01	32	7
IKNSVM	1	100	7
IKSVM	0.01	64	
KSVM	10	1000	

4.3. Comparative Algorithms

In this section we will describe the algorithms that the IDSVM's performance will be compared against. The comparative algorithms are three in total. One is batch leaning based, Kernelized Support Vector Machine (KSVM) and the other two are incremental learning based and they are Incremental Kernelized Support Vector Machine (IKSVM) and Incremental Kernelized Non-Parametric Support Vector Machine (IKNSVM).

4.3.1. KSVM

The KSVM is a powerful ML algorithm used for classification and regression tasks. It is a type of SVM that uses a kernel function to transform

input data into a higher-dimensional space, where it becomes easier to separate different classes using a linear hyperplane. The kernel function is used to measure the similarity between two data points and is used to construct the decision boundary between classes. This allows the SVM to handle complex and non-linear relationships between the input variables.

The work mechanism of KSVMs is explained in a book written by (Burkov, 2019). It basically functions by finding the hyperplane that maximizes the margin between the two classes of data points. The margin is the distance between the hyperplane and the closest data points from each class. This approach helps the SVM to generalize well on unseen data and reduce overfitting. KSVMs are particularly useful when the relationship between input variables is non-linear or when the data is not separable by a linear hyperplane.

There are several types of kernel functions that can be used with SVM, each with its own strengths and weaknesses. Here are some of the commonly used kernel functions:

1. Linear kernel: This kernel function is used for linearly separable data. It creates a straight decision boundary that separates the two classes.
2. Polynomial kernel: This kernel function is used for non-linearly separable data. It maps the input data into a higher-dimensional space using a polynomial function, which makes it easier to separate the two classes.
3. RBF kernel: This kernel function is widely used and can handle both linearly and non-linearly separable data. It maps the input data into a high-dimensional space using a Gaussian function, which creates a circular decision boundary.

4. Sigmoid kernel: This kernel function is used for non-linearly separable data. It maps the input data into a high-dimensional space using a sigmoid function, which creates an S-shaped decision boundary.

Choosing the appropriate kernel function is crucial for the performance of the KSVM algorithm. The choice depends on the nature of the data and the problem at hand (Burkov, 2019).

4.3.2. IKSVM

IKSVM is the same algorithm that is described in subsection 4.3.1. Nevertheless, IKSVM is being trained incrementally instead of being trained on the whole dataset at once, i.e., it does not have access to the full dataset at the training stage. The hyperparameters will be adjusted once a new data point is added. Furthermore, this technique allows the algorithm to preserve the acquired knowledge (Soula et al., 2020).

4.3.3. IKNSVM

The IKNSVM algorithm is a variant of the IKSVM algorithm. Unlike traditional IKSVMs, IKNSVM uses a non-parametric approach to model the decision boundary, which makes them more flexible and better suited to handle complex and non-linear data. In the research conducted by Khan et al. (2014), the KNSVM is described as an algorithm that can address data spread difficulties by incorporating near-global fluctuations from the variance-covariance matrices of the KNDA into the IKSVM optimization problem while considering local data features. Moreover, it applies the kernel method to handle data points that cannot be separated linearly.

4.4. Performance Measures

As stated in section 4.2 above, the algorithms AE, SVM and KNDA went through an iterative process of training and testing until the best performance is achieved. The performance was measured via multiple performance measures. This is to ensure that a holistic understanding of the algorithms' performance is attained. Next, the performance measures are briefly explained along with their mathematical representation.

- **MSE**

A statistical measure that is commonly used to measure the average squared difference between the predicted values and the actual values. MSE is easy to interpret because it has the same units as the data, which makes it easier to compare the performance of different models.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.1}$$

where n is the number of data points, y_i is the i_{th} observed value, $\hat{y}_i$ is the i_{th} predicted value, and Σ is the sum of values from $i = 1\ to\ n$.

- **Accuracy**

Is a measure of how well a classifier is able to correctly predict the outcomes of new, unseen data. It typically measures the proportion of correct predictions out of all predictions made by the algorithm. However, accuracy alone is not always a sufficient measure of algorithm's performance. This is because accuracy can be misleading if the data is imbalanced, meaning that there are many more examples of one class than another. Equation (4.2) for accuracy is expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

- **Precision**

It is a measure of the proportion of true positives among all positive predictions. It is particularly useful when the cost of false positives is high, such as in medical diagnosis or fraud detection. It focuses on the accuracy of positive predictions and is often used in conjunction with recall. High precision means that the model is making few false positive predictions, i.e., it is conservative in labelling instances as positive. It is calculated as per Equation (4.3) below:

$$Precision = \frac{TP}{(TP + FP)} \quad (4.3)$$

- **Recall**

Also known as sensitivity, is a measure of the proportion of true positives over the total number of actual positive instances. It focuses on the ability of the algorithm to identify all positive instances. High recall means that the model is identifying most of the positive instances in the data, i.e., it is inclusive in labelling instances as positive. Recall is useful in cases where false negatives are costly or undesirable. It is calculated as per Equation (4.4) below:

$$Recall = \frac{TP}{(TP + FN)} \quad (4.4)$$

- **False Positive Rate**

Is a measure of the proportion of false positives (incorrectly identified instances) over the total number of actual negative instances. It is commonly used in binary classification problems and is a useful measure of an algorithm's ability to avoid false positives also when negative class is rare or when the cost of false positives is high. The Equation for False Positive Rate is given by (4.5):

$$False\ positive\ Rate = \frac{FP}{(FP + TN)} \tag{4.5}$$

- **Area under the ROC Curve (AUC)**

Is a performance measure that evaluates the overall performance of an algorithm across all possible classification thresholds. It reflects the trade-off between true positive rate and false positive rate and is commonly used to compare different algorithms. AUC is particularly useful when the classes are imbalanced or when the cost of false positives and false negatives is not equal. The receiver operating characteristic (ROC) curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various probability thresholds. AUC-ROC measures the area under the ROC curve and ranges from 0.0 to 1.0, with a value of 0.5 indicating random guessing and a value of 1.0 indicating perfect classification. A higher AUC-ROC value indicates better model performance in terms of distinguishing between positive and negative cases. Equation (4.6) mathematically presents it as:

$$AUC = \int_0^1 TPR d(FPR) \tag{4.6}$$

- **F1 Score**

F1 score is the harmonic mean of precision and recall and is a useful performance measure when there is an uneven class distribution. It balances the importance of precision and recall and is commonly used in binary classification problems where both false positives and false negatives are costly. It ranges from 0 to 1, where a value of 1 indicates perfect precision and recall, and a value of 0 indicates poor performance. Equation (4.7) is illustrating how to calculate F1 score:

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{4.7}$$

- **Execution Time**

A counter is maintained to count the time it took the algorithms to predict the QoE in seconds. The execution time is an essential measure as predicting the QoE is a real-time process.

4.5. Results and Discussion

This research investigates the effectiveness of the proposed novel classifier using a real-world QoE dataset. The effectiveness of such classifier is gauged using the performance measures mentioned in section 4.4. The results will be explained in two subsections. Subsection 4.4.1 will cover the performance results for IDSVM when it was trained using the full datasets. Subsection 4.4.2 will be covering the comparison results for the classifiers when they were trained using a redacted version of the datasets (20% of the original dataset).

4.5.1. Full datasets

Table 4.5 below illustrates the architectures experimented for the AE algorithm. The first and last values represent the input and bottleneck and the number of neurons they contain. The values in between are the hidden layers with the number of neurons they include. Also, the dataset was experimented with no feature selection applied, i.e. the dataset didn't pass through the AE. This

Table (4.5): AE architectures description

Structure	Structure Details
No Dimensionality Reduction Applied "Original Dataset"	No feature selection applied "Dataset didn't pass through AE"
Architecture 1	23-20-18-16
Architecture 2	23-16-10
Architecture 3	23-18-13-8
Architecture 4	23-19-15-11-7

In table 4.6 beneath, AE's performance is expressed in terms of the MSE. The higher the value the lower the AE ability to predict correctly. We can notice that there is no clear relation between the architecture design and the MSE value. However, there is a general tendency to have a lower MSE when the bottleneck size is larger. Nevertheless, it was also noticed that architecture 2 performed better than the other architectures in two out of three datasets. This could be attributable to the smaller number of hidden layer and moderate number of neurons in the hidden layer.

Table (4.6): AE performance

MSE			
	Facebook	**Twitter**	**Instagram**
Architecture 1	7	3	9
Architecture 2	5	4	2
Architecture 3	6	11	24
Architecture 4	59	21	19

Table 4.7 below lists the results of IDSVM performance. Six performance measures were used to understand how well the novel classifier is able to correctly predict the QoE. This is done for all the experimented architectures to understand the impact of the feature selectivity on the classifiers' ability to classify correctly.

Table (4.7): IDSVM performance

Platform/Dataset Structure	**Accuracy**	**Precision**	**Recall**	**FP Rate**	**Area under the ROC Curve**	**F1 Score**
Facebook						
No AE Applied "Original Dataset"	0.7693	0.7681	0.7855	0.2477	0.8194	0.7767
Architecture 1	0.7421	0.7513	0.7402	0.2558	0.7759	0.7457
Architecture 2	0.7436	0.6963	0.8834	0.4024	0.8028	0.7788
Architecture 3	0.7404	0.6945	0.8784	0.4037	0.7969	0.7757
Architecture 4	0.7320	0.6926	0.8588	0.4106	0.7950	0.7651
Twitter						
No AE Applied "Original Dataset"	0.7702	0.7584	0.7926	0.2522	0.8274	0.7751
Architecture 1	0.7420	0.7382	0.7497	0.2657	0.7722	0.7439
Architecture 2	0.7277	0.6741	0.8810	0.4255	0.7905	0.7638
Architecture 3	0.7213	0.6728	0.8612	0.4185	0.7945	0.7554
Architecture 4	0.6938	0.7047	0.7458	0.3583	0.7785	0.6995
Instagram						
No AE Applied "Original Dataset"	0.7667	0.7676	0.7801	0.2473	0.8170	0.7738
Architecture 1	0.7396	0.7555	0.7257	0.2458	0.7765	0.7403
Architecture 2	0.7416	0.6946	0.8831	0.4066	0.7915	0.7776
Architecture 3	0.7396	0.6921	0.8908	0.4256	0.8028	0.7783
Architecture 4	0.7162	0.6779	0.8780	0.4532	0.7678	0.7622

The results per platform will be compared and contrasted using the FP rate, AUC and F1 measures. FP rate acts as an alarm to the algorithm's tolerance for the proportion of incorrect predictions in positive class. At the best-selected threshold, AUC calibrates the trade-off between sensitivity and 1-specificity. It gives the ability to assess the performance of the classifier over its entire operating range. As for the F1 score, it is able to reveal the actual algorithm's performance when the dataset is imbalanced. Hence, aforementioned measures are chosen to do the comparisons.

Table 4.8 below compares the performance measures for the original dataset and four different AE architectures (Architecture 1, Architecture 2, Architecture 3, and Architecture 4) for Facebook dataset. The performance measures are FP Rate, Area under the ROC Curve, and F1 Score.

Table (4.8): IDSVM performance for Facebook dataset

Facebook			
Dataset Structure	**FP Rate**	**Area under the ROC Curve**	**F1 Score**
No AE Applied "Original Dataset"	0.2477	0.8194	0.7767
Architecture 1	0.2558	0.7759	0.7457
Architecture 2	0.4024	0.8028	0.7788
Architecture 3	0.4037	0.7969	0.7757
Architecture 4	0.4106	0.7950	0.7651

A lower FP rate indicates better performance. In this case, the original dataset has the lowest FP rate of 0.2477, followed by Architecture 1 with an FP rate of 0.2558. The remaining architectures have higher FP rates, with Architecture 4 having the highest FP rate of 0.4106. A higher AUC implies superior performance. Hence, the original dataset has the highest AUC of 0.8194, followed closely by Architecture 2 with an AUC of 0.8028. The remaining architectures have lower AUC values, with

Architecture 4 having the lowest AUC of 0.7950. A higher F1 score suggests better performance. Architecture 2 has the highest F1 score of 0.7788, followed closely by the original dataset with an F1 score of 0.7767. The remaining architectures have lower F1 scores, with Architecture 4 having the lowest F1 score of 0.7651. In summary, the performance measures values for the different architectures are generally lower compared to the original dataset, but Architecture 2 has the highest F1 score.

Table 4.9 below compares the performance measures for the original dataset and four different AE architectures (Architecture 1, Architecture 2, Architecture 3, and Architecture 4) for Twitter dataset. The performance measures are FP Rate, Area under the ROC Curve, and F1 Score.

Table (4.9): IDSVM performance for Twitter dataset

Twitter			
Dataset Structure	**FP Rate**	**Area under the ROC Curve**	**F1 Score**
No AE Applied "Original Dataset"	0.2522	0.8274	0.7751
Architecture 1	0.2657	0.7722	0.7439
Architecture 2	0.4255	0.7905	0.7638
Architecture 3	0.4185	0.7945	0.7554
Architecture 4	0.3583	0.7785	0.6995

Comparing the values in the Table 4.9, we can see that the original dataset has the lowest FP rate (0.2522) and highest AUC (0.8274) and F1 score (0.7751) compared to all four architectures. Among the architectures, Architecture 1 has the second-best performance for the FP rate (0.2657). Architecture 2 has the highest FP rate (0.4255) among all architectures, indicating that it incorrectly classified more negative samples as positive. However, it has a relatively high AUC (0.7905) and F1 score (0.7638) compared to some other architectures. Architecture 3 has a similar AUC

value (0.7945) as Architecture 2, but a slightly lower FP rate (0.4185) and F1 score (0.7554). Architecture 4 has a relatively low FP rate (0.3583) compared to Architecture 2 and 3, but it has the lowest AUC (0.7785) and F1 score (0.6995) among all architectures, indicating that it has relatively poor performance in distinguishing between positive and negative samples.

Table 4.10 below provides the performance measures for the original dataset and four different AE architectures (Architecture 1, Architecture 2, Architecture 3, and Architecture 4) for Instagram dataset. The performance measures are FP Rate, Area under the ROC Curve, and F1 Score.

Table (4.10): IDSVM performance for Instagram dataset

Instagram			
Dataset Structure	**FP Rate**	**Area under the ROC Curve**	**F1 Score**
No AE Applied "Original Dataset"	0.2473	0.8170	0.7738
Architecture 1	0.2458	0.7765	0.7403
Architecture 2	0.4066	0.7915	0.7776
Architecture 3	0.4256	0.8028	0.7783
Architecture 4	0.4532	0.7678	0.7622

Looking at the FP rate column, we can observe that Architecture 1 has the lowest value of 0.2458, followed by the original dataset with 0.2473. Architecture 3 has a considerably higher value of 0.4256, followed by Architecture 2 with 0.4066, and Architecture 4 with the highest value of 0.4532. Moving to the area under the ROC curve, we can observe that the original dataset has the highest value of 0.817, followed by Architecture 3 with 0.8028, Architecture 2 with 0.7915, Architecture 1 with 0.7765, and Architecture 4 with the lowest value of 0.7678. Looking at the F1 score column, we can observe that Architecture 3 has the highest value of 0.7783, followed by Architecture 2 with 0.7776, the original dataset with 0.7738, Architecture 4 with 0.7622, and Architecture 1 with the lowest value of

0.7403. Architecture 1 has a low FP rate but a lower area under the ROC curve and F1 score. Architecture 4 shows relatively poor performance across all performance measures.

From Tables 4.8, 4.9 and 4.10, we can observe that the performance generally differs slightly between the original datasets and the 4 architectures. In other words, the gap in performance is not dramatically huge. The fluctuation in the performance is considered limited. This suggests that the AE has successfully remove the less important attributes and only maintained the ones that are most relevant to the output. The fluctuation in the results among the tested scenarios is minimal for the AUC and F1 score measure, however this was not the case for the FR rate measure, were architecture 2,3 and 4 scored noticeably lower than the other scenarios. This suggests that the classifier's ability to identify the positive classes is affected by the number of attributes fed into the classifier.

Fig. 4.1 below presents a comparison from a different angle. It illustrates the IDSVM's performance per social media platform for Architecture 1. The comparison is done specifically for this architecture because we wish to exhibit the impact of the minimal dimensionality reduction on the classifier's performance and because this architecture constantly presented low FP rates compared to other architectures.

Looking at the FP rate, we can observe that Instagram has the lowest value of 0.2458, followed by Facebook with 0.2558 and Twitter with 0.2657. This suggests that the classifier performs best on Instagram when it comes to reducing false positives. Moving to the area under the ROC curve, we can observe that the values for the three datasets are relatively close. Instagram has the highest value of 0.7765, followed by Facebook with 0.7759 and Twitter with 0.7722. This suggests that the classifier performs slightly better on Instagram and Facebook in terms of overall performance, as indicated by the area under the ROC curve. As for the F1

score, we can observe that Facebook has the highest value of 0.7457, followed by Twitter with 0.7439 and Instagram with 0.7403. This suggests that the classifier performs best on Facebook when it comes to balancing precision and recall, as indicated by the F1 score.

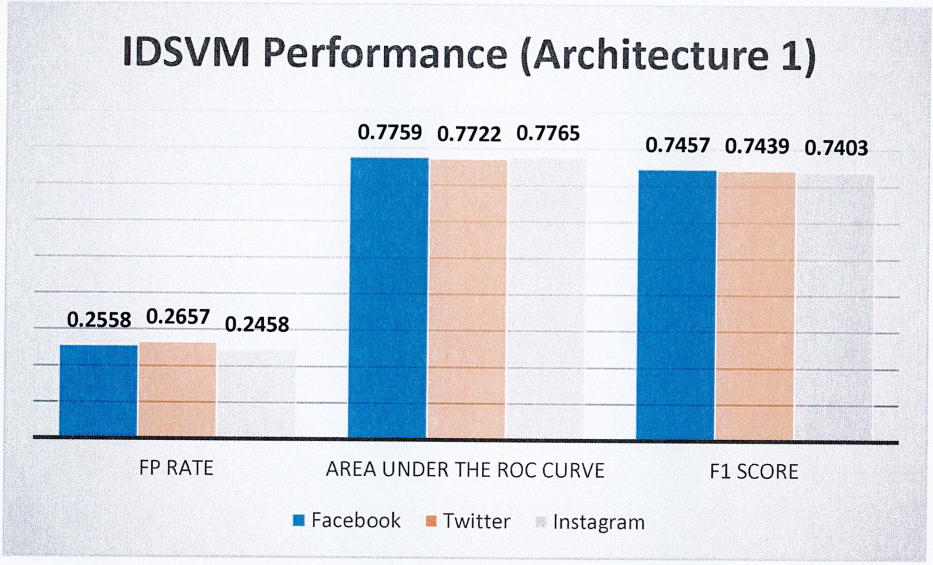


Fig. 4.1: IDSVM performance for architecture 1

It can be concluded that the classifier performs differently on the three datasets in terms of the evaluated metrics. Instagram has the lowest FP rate, while Facebook has the highest F1 score. Instagram and Facebook have similar performance in terms of the area under the ROC curve, but slightly better than Twitter.

Table 4.11 below presents the time range recorded to train the IDSVM per platform.

Table (4.11): Training time per social media platform

Social Media Platform	Training Time (in seconds)
	Minimum - Maximum
Facebook	206-267
Twitter	151-212
Instagram	199-259

Comparing the training time across the platforms, we can see that Twitter has the lowest minimum training time of 151 seconds, while Facebook has the highest minimum training time of 206 seconds. However, the maximum training time for Facebook and Instagram are close at 267 and 259 seconds, respectively.

It is also interesting to note that the range of training time for each platform is relatively narrow (61 seconds). This suggests that the training time for the IDSVM classifier is fairly consistent across the different social media platforms.

4.5.2. Redacted datasets

A redacted versions of the original datasets were used for the sole purpose of comparing the performance of the classifiers. It was not meant to show case the top performance of the classifiers. Following the redacted datasets approach was dictated by computing resources constraints. The trainings datasets went under dimensionality reduction to match architecture 1, i.e., only 16 attributes retained from the 23 attributes.

Table 4.12 below illustrates the performance per classifier per dataset. Area under the ROC curve performance measure is adopted to evaluate the performance.

Table (4.12): Performance per classifier per dataset

Area under the ROC Curve			
	Facebook	**Twitter**	**Instagram**
IDSVM	0.6079	0.6744	0.6169
IKNSVM	0.6093	0.5867	0.6033
IKSVM	0.5496	0.5433	0.5493
KSVM	0.5518	0.5472	0.5122

From Table 4.12, we can observe the following about the classifiers' performance on each dataset based on the area under the ROC curve performance measure:

Facebook dataset:

IDSVM and IKNSVM classifiers have comparable performance with scores of 0.6079 and 0.6093, respectively, while the performance of IKSVM and KSVM is lower, with scores of 0.5496 and 0.5518, respectively. The difference in performance between the top two classifiers (IDSVM and IKNSVM) and the bottom two (IKSVM and KSVM) is relatively large, indicating that IDSVM and IKNSVM are more effective at separating the positive and negative instances in this dataset.

Twitter dataset:

IDSVM is the best performer on this dataset with a score of 0.6744. The performance of the other three classifiers (IKNSVM, IKSVM, and KSVM) is much lower, with scores ranging from 0.5433 to 0.5472. The difference in performance between IDSVM and the other classifiers is quite significant, suggesting that IDSVM is particularly effective at separating positive and negative instances in this dataset.

Instagram dataset:

IDSVM and IKNSVM have comparable performance on this dataset, with scores of 0.6169 and 0.6033, respectively. The performance of IKSVM is slightly lower than that of the top two classifiers, with a score of 0.5493. KSVM has the lowest performance score of all classifiers on this dataset, with a score of 0.5122. The difference in performance between the top two classifiers and the bottom two classifiers is relatively large, indicating that IDSVM and IKNSVM are more effective at separating the positive and negative instances in this dataset than IKSVM and KSVM.

In summary, the performance of each classifier varies depending on the dataset. IDSVM and IKNSVM are the most effective classifiers overall, while IKSVM and KSVM are less effective. However, the performance rankings of the classifiers differ on each dataset. On the Facebook dataset, IDSVM and IKNSVM are the best performers, while IDSVM is the best performer on the Twitter dataset. On the Instagram dataset, IDSVM and IKNSVM have comparable performance. Overall, IDSVM has performed slightly better.

Table 4.13 below shows the performance per classifier per dataset in terms of the consumed training time.

Table (4.13): Training time per classifier per dataset

Training time in seconds			
	Facebook	**Twitter**	**Instagram**
IDSVM	14.72	14.66	14.72
IKNSVM	619.81	364.81	971.53
IKSVM	11.67	11.69	11.66
KSVM	274.69	275.10	287.26

Facebook dataset:

Here we can see that IKSVM has the lowest training time (11.67 seconds), followed by IDSVM (14.72 seconds) and KSVM (274.69 seconds). On the other hand, IKNSVM has a significantly higher training time (619.81 seconds) than the other classifiers.

Twitter dataset:

Again, IKSVM has the lowest training time (11.69 seconds), followed by IDSVM (14.66 seconds) and KSVM (275.10 seconds). However, the training time for IKNSVM is still significantly higher (364.81 seconds) than the other classifiers.

Instagram dataset:

Similar to the previous datasets, IKSVM has the lowest training time (11.66 seconds), followed by IDSVM (14.72 seconds) and KSVM (287.26 seconds). However, the training time for IKNSVM is again significantly higher (971.53 seconds) than the other classifiers.

Overall, we can see that IKSVM has the lowest training time among all classifiers for all three datasets as it learns incrementally. IDSVM has the second-lowest training time due to the production of the variance covariance matrices, while KSVM has a moderate training time as it learns in batch manner. IKNSVM, on the other hand, has a significantly higher training time than the other classifiers for all three datasets since it calculates the variance covariance matrices in the feature space rather than the input space which consumes more time to resolve the complex higher dimensional calculation.

Chapter Five

Conclusion and Recommendation

5.1. Conclusion

5.2. Recommendation

Chapter Five
Conclusion and Recommendation

5.1. Conclusion

Based on the conducted literature review, the value that this research is adding is that it made use of SVM and KNDA strengths to predict the QoE for web browsing of top social media platforms. Moreover, the algorithms were trained incrementally to cater for the dynamic nature of the data generation in telecom networks. This novel combination of algorithms was called IDSVM. The IDSVM was trained using real life data collected via crowdsourcing tool. The data contained network performance indicators along with the timing and location of the tests.

The feature selection worked quite well. The AE managed to eliminate the attributes that are not much related to the output (QoE). Such conclusion is evidenced by the minimal difference in performance of the IDSVM between the original dataset and datasets with eliminated attributes. The IDSVM's was evaluated against three algorithms. Two of which are incremental based algorithms. Using the area under ROC curve performance measure, IDSVM (0.6079) and IKNSVM (0.6093) are the best performers for Facebook dataset. However, IDSVM is the best performer on the Twitter (0.6744) and Instagram (0.6169) datasets. In terms of the time that each classifier consumed to train, IKSVM has the lowest training time among all classifiers for all three datasets. IDSVM has the second-lowest training time. It took 25% more time to train than IKSVM due to the complexities of producing the matrices. The other two classifiers performed worse than IDSVM by multiple factors.

5.2. Recommendation

In this research, the adopted and evaluated algorithms were SVM based. Moreover, the output was binary in nature, and it was determined depending on pre-set thresholds and "if" conditions. In addition, the attributes within the datasets were limited to the technical indicators. The aforementioned can be seen as potential areas for enhancements.

Therefore, for future work, it is recommended to deploy non SVM based algorithms for a wider comparison and understanding of the algorithms' capabilities. Also, it is recommended to use the highly adopted MOS scheme to rate the QoE. For this subjective rating, human ratings will be required. Nevertheless, it will give better understanding of the perceived QoE. In addition, considering the non-technical aspect such as age, gender and income is highly recommended as it will assist the TSP to customize the provided services to meet the consumers' expectations. Moreover, revisiting the established thresholds is recommended to keep up with the latest best practices. Finally, it is recommended to attempt to enhance the performance of the algorithms by deploying optimization techniques such as the genetic algorithm.

References

Aroussi, S., Bouabana-Tebibel, T., & Mellouk, A. (2012). Empirical QoE/QoS correlation model based on multiple parameters for VoD flows. *GLOBECOM - IEEE Global Telecommunications Conference*, 1963–1968. https://doi.org/10.1109/GLOCOM.2012.6503403

Aroussi, S., & Mellouk, A. (2016, June 27). Statistical evaluation for quality of experience prediction based on quality of service parameters. *2016 23rd International Conference on Telecommunications, ICT 2016*. https://doi.org/10.1109/ICT.2016.7500364

Bai, Y. (2022). RELU-Function and Derived Function Review. *SHS Web of Conferences, 144*, 02006. https://doi.org/10.1051/shsconf/202214402006

Bank, D., Koenigstein, N., & Giryes, R. (2020). *Autoencoders*. http://arxiv.org/abs/2003.05991

Begluk, T., Husic, J. B., & Barakovic, S. (2018). Machine learning-based QoE prediction for video streaming over LTE network. *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018 - Proceedings, 2018-January*, 1–5. https://doi.org/10.1109/INFOTEH.2018.8345519

Ben Youssef, Y., Afif, M., Ksantini, R., & Tabbane, S. (2018). A novel online QoE prediction model based on multiclass incremental support vector machine. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 2018-May*, 334–341. https://doi.org/10.1109/AINA.2018.00058

Burkov, A. (2019). *The Hundred-page Machine Learning Book*.

Casas, P., D'Alconzo, A., Wamser, F., Seufert, M., Gardlo, B., Schwind, A., Tran-Gia, P., & Schatz, R. (2017, June 30). Predicting QoE in cellular networks using machine learning and in-smartphone measurements. *2017 9th International Conference on Quality of Multimedia Experience, QoMEX 2017*. https://doi.org/10.1109/QoMEX.2017.7965687

Clark, B. (2020, March 16). *How to Improve Customer Retention in the Telecom Industry*. Acquire. https://acquire.io/blog/improve-customer-retention-in-telecom-industry

Da Hora, D., Asrese, A. S., Christophides, V., Teixeira, R., Rossi, D., Neves. (2018). *Narrowing the gap between QoS metrics and Web QoE using Above-the-fold metrics*. https://hal.inria.fr/hal-01677260

Davison, A. C. (2003). *Statistical Models*. Cambridge University Press.

Delong, L., & Kozak, A. (2022). *The use of autoencoders for training neural networks with mixed categorical and numerical features*. https://ssrn.com/abstract=3952470

Diaz-Aviles, E., Pinelli, F., Lynch, K., Nabi, Z., Gkoufas, Y., Bouillet, E., Calabrese, F., Coughlan, E., Holland, P., & Salzwedel, J. (2015). *Towards Real-time Customer Experience Prediction for Telecommunication Operators*. http://arxiv.org/abs/1508.02884

Efazati, S., Amerimehr, M. H., Montazeri, A. M., Sabagh, M. R., Alibeigi, M., & Ayazi, F. (2020). *AUT Journal of Electrical Engineering Novel Methods For Determining QoS Parameters and Thresholds in End user's Service Level Agreement*. *52*(2), 205–216. https://doi.org/10.22060/eej.2020.17659.5323

Elwerghemmi, R., Heni, M., Ksantini, R., & Bouallegue, R. (2019). Online QoE Prediction Model Based on Stacked Multiclass Incremental Support Vector Machine. *International Conference on Modeling, Simulation, and Applied Optimization*. https://doi.org/10.1109/icmsao.2019.8880302

Fiedler, M., Hossfeld, T., & Tran-Gia, P. (2010). A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, *24*(2), 36–41. https://doi.org/10.1109/MNET.2010.5430142

Flores, S. (2019, April 15). *Variational autoencoders are beautiful*. Compthree. https://www.compthree.com/blog/autoencoder/

Frnda, J., Nedoma, J., Vanus, J., & Martinek, R. (2019). A hybrid QoS-QoE estimation system for IPTV service. *Electronics (Switzerland)*, *8*(5). https://doi.org/10.3390/electronics8050585

Gao, Y., Wei, X., Zhuang, W., Huang, R., & Diao, M. (2017). QoE prediction for IPTV based on BP-adaboost neural networks. *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, 32–37. https://doi.org/10.1109/IWCMC.2017.7986258

Gatara, M. C., & Mzyece, M. (2022). From QoS to QoE plus QoT in Beyond 5G Networks. *Journal of Mobile Multimedia*, *19*(1), 135–146. https://doi.org/10.13052/jmm1550-4646.1917

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Hu, Z. G., Yan, H. R., Yan, T., Geng, H. J., & Liu, G. Q. (2020). Evaluating QoE in VoIP networks with QoS mapping and machine learning algorithms. *Neurocomputing*, *386*, 63–83. https://doi.org/10.1016/j.neucom.2019.12.072

Huang, P. J. (2015). *Classification of Imbalanced Data Using Synthetic Over-Sampling Techniques* [Master's thesis, University of California]. https://escholarship.org/content/qt72w743h7/qt72w743h7_noSplash_bd589490abd5ac433d09090e910cb967.pdf

International Chamber of Commerce. (2007). *Telecoms liberalization: An international business guide for policymakers* (2nd ed.). https://iccwbo.org/content/uploads/sites/3/2007/10/ICC-EBITT-Commission-Telecoms-Liberalization-An-international-business-guide-for-policymakers.pdf

International Telecommunications Union. (2022, January 19). *The relation between quality of service and quality of experience.*

https://digitalregulation.org/the-relation-between-quality-of-service-and-quality-of-experience/

Kefi-Fatteh, T., Ksantini, R., Kaâniche, M. B., & Bouhoula, A. (2019). A novel incremental one-class support vector machine based on low variance direction. *Pattern Recognition, 91*, 308–321. https://doi.org/10.1016/j.patcog.2019.02.027

Khan, N. M., Ksantini, R., Ahmad, I. S., & Guan, L. (2014). SN-SVM: a sparse nonparametric support vector machine classifier. *Signal, Image and Video Processing, 8*(8), 1625–1637. https://doi.org/10.1007/s11760-012-0404-3

Kumar, A. (2022, December 5). *Difference between online & batch learning.* Vitalflux. https://vitalflux.com/difference-between-online-batch-learning/

Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., & Zhang, C. (2019). *CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks.* http://arxiv.org/abs/1904.09483

Littledata. (2022, September 28). *What is the average time before full mobile page load?.* https://lp.littledata.io/average/time-before-full-mobile-page-load

Losing, V., Hammer, B., & Wersing, H. (2016). Choosing the Best Algorithm for an Incremental On-line Learning Task. *The European Symposium on Artificial Neural Networks.* http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2016-71.pdf

Piamrat, K., Viho, C., Ksentini, A., & Bonnin, J. M. (2009). Quality of experience measurements for video streaming over wireless networks. *ITNG 2009 - 6th International Conference on Information Technology: New Generations*, 1184–1189. https://doi.org/10.1109/ITNG.2009.121

Rubino, G., & Varela, M. (2004). A new approach for the prediction of end-to-end performance of multimedia streams. *First International Conference on the Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings.* https://doi.org/10.1109/qest.2004.1348025

Salhi, M., Ksantini, R., & Zouari, B. (2022). A real-time image-centric transfer function design based on incremental classification. *Journal of Real-Time Image Processing, 19*(1), 185–203. https://doi.org/10.1007/s11554-021-01176-x

Saverimoutou, A., Mathieu, B., & Vaton, S. (2019). A 6-month analysis of factors impacting web browsing quality for QoE prediction. *Computer Networks, 164*. https://doi.org/10.1016/j.comnet.2019.106905

ScreenBeam (n.d.). *Wi-Fi signal strength: what is a good signal?*. Retrieved February 18, 2023, from https://www.screenbeam.com/wifihelp/wifibooster/wi-fi-signal-strength-what-is-a-good-signal/#:~:text=%2D60%20dBm%3A%20This%20is%20a,not%20a%20strong%20signal%20strength.

Sematext. (n.d.). *What is a good dns lookup time?*. Retrieved February 18, 2022, https://sematext.com/glossary/dns-lookup-time/#:~:text=The%20average%20DNS%20lookup%20time,is%20generally%20considered%20very%20good.

Shan, Y., Wang, J., Xie, X., Meng, L., & Kuang, J. (2018). Non-intrusive Speech Quality Assessment Using Deep Belief Network and Backpropagation Neural Network. *International Symposium on Chinese Spoken Language Processing*. https://doi.org/10.1109/iscslp.2018.8706696

Sharma, A. (2022, March 22). *Pros and cons of commonly used statistical models*. Medium. https://medium.com/@aruneema98/pros-and-cons-of-commonly-used-statistical-models-1f868280bbf8

Song, E., Pan, T., Fu, Q., Zhang, R., Jia, C., Cao, W., & Huang, T. (2020). Threshold-oblivious on-line web QoE assessment using neural network-based regression model. *IET Communications, 14*(12), 2018–2026. https://doi.org/10.1049/iet-com.2019.1229

Soula, A., Tbarki, K., Ksantini, R., Said, S. Ben, & Lachiri, Z. (2020). A novel incremental Kernel Nonparametric SVM model (iKN-SVM) for data classification: An application to face detection. *Engineering Applications of Artificial Intelligence, 89*. https://doi.org/10.1016/j.engappai.2019.103468

Tao, X., Duan, Y., Xu, M., Meng, Z., & Lu, J. (2019). Learning QoE of Mobile Video Transmission with Deep Neural Network: A Data-Driven Approach. *IEEE Journal on Selected Areas in Communications, 37*(6), 1337–1348. https://doi.org/10.1109/JSAC.2019.2904359

Telecommunication Regulatory Authority. (n.d.). *QoS crowdsourcing maps*. Retrieved February 3, 2022, from https://speedtest.tra.org.bh/coverage-maps/

Telecommunications Standardization Sector of International Telecommunications Union. (2020). *Overall network operation, telephone service, service operation and human factors: Quality of telecommunication services: concepts, models, objectives and dependability planning – Models for telecommunication services*. In (Rec. ITU-T E.812). International Telecommunications Union. https://www.itu.int/rec/T-REC-E.812/en

Telecommunications Standardization Sector of International Telecommunications Union. (2019). *Telephone transmission quality, telephone installations, local line networks: Vocabulary and effects of transmission parameters on customer opinion of transmission quality*. (Rec. ITU-T P.10/G.100). International Telecommunications Union. https://www.itu.int/rec/T-REC-P.10

Telecommunications Standardization Sector of International Telecommunications Union. (2016). *Terminals and subjective and objective assessment methods: Methods for objective and subjective assessment of speech and video quality*. (Rec. ITU-T P.800.1). International Telecommunications Union. https://www.itu.int/rec/T-REC-P.800.1

Telecommunications Standardization Sector of International Telecommunications Union. (2008). *Overall network operation, telephone service, service operation and human factors: quality of*

telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services. (Rec. ITU-T E.800). International Telecommunications Union. https://www.itu.int/rec/T-REC-E.800-200809-I

Tutela. (2021, September 1). *Introducing Consistent Quality.* https://www.tutela.com/consistent-quality

Teltonika. (2022, March 4). *Mobile signal strength recommendations.* https://wiki.teltonika-networks.com/view/Mobile_Signal_Strength_Recommendations

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. *IEEE Access, 7,* 60134–60149. https://doi.org/10.1109/ACCESS.2019.2914999

Vasilev, V., Leguay, J., Paris, S., Maggi, L., & Debbah, M. (2018). Predicting QoE Factors with Machine Learning. *IEEE International Conference on Communications, 2018-May.* https://doi.org/10.1109/ICC.2018.8422609

Zack, (2020, November 30). *How to normalize data between 0 and 100.* Statology. https://www.statology.org/normalize-data-between-0-and-100/#:~:text=to%20be%201%2C000%3A,To%20normalize%20the%20first%20value%20of%2012%2C%20we%20would%20apply,%E2%80%93%2012)%20*%20100%20%3D%200

Zafar, S., Jangsher, S., Bouachir, O., Aloqaily, M., & Ben Othman, J. (2019). QoS enhancement with deep learning-based interference prediction in mobile IoT. *Computer Communications, 148,* 86–97. https://doi.org/10.1016/j.comcom.2019.09.010

Zhang, A., Lipton, Z., Li, M., & Smola, A. J. (2022a). *Dive Into Deep Learning* (1st ed.). The authors. https://d2l.ai/d2l-en.pdf

Zhang, X., Li, S., & Hu, F. (2022b). Convolutional Recurrent Neural Networks with Attention Mechanism for Streaming QoE Prediction. *2022 6th*

International Conference on Communication and Information Systems, ICCIS 2022, 104–108. https://doi.org/10.1109/ICCIS56375.2022.9998164

Zhang, X., & Wang, J. (2018). Joint heterogeneous statistical-QoS/QoE provisionings for edge-computing based WiFi offloading over 5G mobile wireless networks. *2018 52nd Annual Conference on Information Sciences and Systems, CISS 2018*, 1–6. https://doi.org/10.1109/CISS.2018.8362265

Zhu, S., Dong, Y., & Xu, C. (2020). A Statistical QoE-QoS Model of Video Streaming Services. *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*. https://doi.org/10.1145/3379247.3379252

الملخص

يسعى مزودي خدمات الاتصالات لتقديم خدمات أفضل وتلبية طلبات عملائهم المتزايدة باستمرار، وعلى الرغم من جهودهم إلا أنهم يواجهون تحديات كبيرة في ضمان جودة التجربة والتنبؤ فيها. ففي سوق شديد التنافسية، يكون لدى العملاء توقعات عالية بأن يكون الاتصال موثوق وسريع. ويعد ضمان جودة التجربة العالية أمراً في بالغ الأهمية للاحتفاظ بالعملاء والحفاظ على صورة إيجابية للعلامة التجارية. ومع ذلك، فإن التنبؤ بجودة التجربة مهمة معقدة تتضمنها عوامل مختلفة. ولحسن الحظ، ظهر تعلم الآلة كأداة فعالة للتنبؤ بجودة التجربة لمساعدة مزودي خدمات الاتصالات على أن يلبو ويفوقو توقعات عملائهم. لذلك، وجد أنه من الممكن إجراء بحث ضمن مجال التنبؤ بجودة التجربة خاصة مع محدودية البحوث في هذا المجال. وبالتالي، كان لدينا الدافع لتجربة نموذج مبتكر يتكون من التحليل التمييزي Discriminant Analysis ودعم آلة المتجه Support Vector Machine.حيث أن هذا النموذج أن يجمع بين كل المتغيرات القريبة والبعيدة التي يوفرها الحد الأقصى للهامش والقائمة على توجهات التمييز. بالإضافة إلى ذلك، فإنه يستخدم مساحة الإدخال بدلا من فراغ المتجهات لإنشاء مصفوفة التباين والتغايير. و على العكس من النماذج الأخرى، فقد تم تدريب هذا النموذج على البيانات بشكل تدريجي بدل تدريبه على البيانات كلها في آن واحد، وهو ما يناسب طبيعة حاجة مزودي خدمات الاتصالات للتنبؤ بجودة التجربة بشكل آني.

تم الحصول على بيانات التدريب من هيئة تنظيم الاتصالات البحرينية. هذا البحث هو أول من استخدم مجموعة البيانات هذه واستعان بالنموذج الجديد في مجال التنبؤ بجودة التجربة. احتوت مجموعة البيانات هذه على البيانات الفنية حول تجربة الوصول إلى صفحة الويب الخاصة ب Facebookو Twitterو .Instagramمجموعة البيانات قد مرت بمراحل من الإعداد، تبدأ بنقية البيانات وإدراج وموازنة المخرجات وانتهت باختيار السمات الاكثر علاقة بالمخرجات عن طريق الـ Autoencoder.

تم تقييم النموذج المبتكر باستخدام المنطقة تحت منحنى ROC كما أنه تم اعتبار الوقت المستهلك في التدريب من ضمن أدوات التقييم. تفوق النموذج المبتكر في قدرته على التنبؤ الصحيح بالمخرجات بالمقارنة مع نماذج أخرى لبيانات Twitter و Instagram كما حل بالمركز الثاني من ناحية النموذج الأقل استهلاكاً للوقت المطلوب للتدريب .

بشكل عام، يمكن اعتبار أداء النموذج المبتكر جيد نسبياً كما يمكن اعتباره نموذج سريع الأداء. وبالتالي، يمكن استخدام النموذج المبتكر للتطبيقات التي تتطلب سرعة آنية في اتخاذ القرار.

جامعة البحرين

كلية العلوم **قسم برامج الدراسات العليا**

تقنية تعلم عميق تزايدي مبتكرة لتفسير جودة الخدمة وتوقع جودة الاستخدام لأهم منصات التواصل الاجتماعي

إطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير
في علوم وتحليل البيانات الضخمة

إعداد

عيسى ياسر عيسى عجلان الحدي

202100022

إشراف

د . رياض بن محمد قسنطيني

أستاذ مشارك

جامعة البحرين

مملكة البحرين

مارس 2023