

Spring 2024

Applications of Predictive and Generative AI Algorithms: Regression Modeling, Customized Large Language Models, and Text-to-Image Generative Diffusion Models

Suhaima Jamal

Follow this and additional works at: <https://digitalcommons.georgiasouthern.edu/etd>

 Part of the [Computer Engineering Commons](#)

Recommended Citation

Jamal, Suhaima, "Applications of Predictive and Generative AI Algorithms: Regression Modeling, Customized Large Language Models, and Text-to-Image Generative Diffusion Models" (2024). *Electronic Theses and Dissertations*. 2715.
<https://digitalcommons.georgiasouthern.edu/etd/2715>

This thesis (open access) is brought to you for free and open access by the Jack N. Averitt College of Graduate Studies at Georgia Southern Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Georgia Southern Commons. For more information, please contact digitalcommons@georgiasouthern.edu.

APPLICATIONS OF PREDICTIVE AND GENERATIVE AI ALGORITHMS: REGRESSION
MODELING, CUSTOMIZED LARGE LANGUAGE MODELS, AND TEXT-TO-IMAGE
GENERATIVE DIFFUSION MODELS

by

SUHAIMA JAMAL

Under the Direction of Hayden Wimmer

ABSTRACT

The integration of Machine Learning (ML) and Artificial Intelligence (AI) algorithms has radically changed predictive modeling and classification tasks, enhancing a multitude of domains with unprecedented analytical capabilities. Predictive modeling leverages ML and AI to forecast future trends or behaviors based on historical data, while classification tasks categorize data into distinct classes, from email filtering to medical diagnosis. Concurrently, text-to-image generation has emerged as a transformative potential, allowing visual content creation directly from textual descriptions. These advancements are pivotal in design, art, entertainment, and visual communication, as well as enhancing creativity and productivity. This work explores three significant studies in ML and AI research, focusing on predictive and classification solutions on cloud platforms. First, a study evaluates regression-type ML models across cloud platforms, offering critical insights for optimizing models and deployment strategies. Second, research on customizing large language models for email classification addresses cybersecurity concerns, bolstering email security measures. Moreover, this work demonstrates how LLMs can be customized via training existing models on new data. Finally, investigation into text-to-image generation diffusion models highlights the evolving landscape of AI-driven visual content generation while informing future advancements and applications. Together, these studies advance the capabilities and applications of ML and AI technologies, addressing real-world challenges and driving innovation.

INDEXED WORDS: Artificial intelligence, Machine learning, Large language models, Text-to-image generation, Diffusion models, Deep learning, Cloud computing, Microsoft Azure, AWS, GCP, DistilBERT, RoBERTA, Fine-tuning, Model optimization

APPLICATIONS OF PREDICTIVE AND GENERATIVE AI ALGORITHMS: REGRESSION
MODELING, CUSTOMIZED LARGE LANGUAGE MODELS, AND TEXT-TO-IMAGE
GENERATIVE DIFFUSION MODELS

by

SUHAIMA JAMAL

B.Sc., Computer Science & Engineering, Chittagong University of Engineering & Technology,
Bangladesh, 2019

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

STATESBORO, GEORGIA

© 2024

SUHAIMA JAMAL

All Rights Reserved

APPLICATIONS OF PREDICTIVE AND GENERATIVE AI ALGORITHMS: REGRESSION
MODELING, CUSTOMIZED LARGE LANGUAGE MODELS, AND TEXT-TO-IMAGE
GENERATIVE DIFFUSION MODELS

by

SUHAIMA JAMAL

Major Professor:

Hayden Wimmer

Committee:

Meenalosini Vimal Cruz

Kim Jongyeop Kim

Electronic Version Approved:

May 2024

DEDICATION

I am grateful to the Almighty for blessing me with sound health and the intellectual capacity to pursue my degree and complete this thesis work. I dedicate this to my parents, whose constant support and encouragement have been my guiding light throughout this journey, and to my amazing husband for being consistently supportive during times of good and bad.

ACKNOWLEDGMENTS

I extend my heartfelt gratitude to Dr. Hayden Wimmer, my thesis advisor, for his invaluable guidance. Since the very first day of my graduate class until this day when I am submitting my final thesis copy, he has demonstrated passion and dedication in supporting me throughout the course of this study. From tackling complex research methodology designs to swiftly resolving basic Word formatting issues, he consistently offers prompt support whenever needed. I am beyond thankful for all of these!

I am deeply grateful to each of my thesis committee members for their constructive feedback and valuable contributions to this research endeavor. Especially, I would like to express my sincere appreciation to Dr. Meenalosini Vimal Cruz for her continuing understanding and assistance as my graduate assistant supervisor during my master's program.

Finally, my family has always been my pillars of support, offering encouragement and motivation. Their belief in my abilities has been a constant source of strength throughout my academic journey.

Thank you all!

TABLE OF CONTENTS

DEDICATION	1
ACKNOWLEDGMENTS	2
TABLE OF CONTENTS	3
LIST OF FIGURES	6
LIST OF TABLES	8
CHAPTER 1: INTRODUCTION	9
CHAPTER 2: LITERATURE REVIEW	12
2.1 STUDY A – LITERATURE REVIEW: REGRESSION MODELS EVALUATION ON CLOUD PLATFORMS: AWS VS AZURE VS GCP	12
2.1.1. Supervised ML Algorithms’ Performance Comparison	12
2.1.2. ML Model Evaluation on Open-source Data Mining Tools	13
2.1.3. Cloud Vendors’ Comparison: Services and Design Taxonomies	14
2.2 STUDY B – LITERATURE REVIEW: An Improved Transformer-based Model for Detecting Phishing, Spam and Ham Emails: A Large Language Model Approach	15
2.2.1. Machine Learning and Deep Learning-based Methods	15
2.2.2 Transformer Model-based Approaches	16
2.3 STUDY C – LITERATURE REVIEW: PERCEPTION AND EVALUATION OF TEXT-TO- IMAGE GENERATIVE AI MODELS: A COMPARATIVE STUDY OF DALL-E, GOOGLE IMAGEN, GROK, AND STABLE DIFFUSION	17
CHAPTER 3: STUDY A- REGRESSION MODELS EVALUATION ON CLOUD PLATFORMS: AWS VS AZURE VS GCP	20
3.1. Introduction	20
3.2. Methods	22
3.2.1. Datasets	22
3.2.2. Cloud Platforms	25
3.2.3. Algorithm	25
3.2.4. Procedure	25
3.3. Experimental Results	29

3.3.1. R Squared Value or Coefficient of Determination	29
3.3.2. Error Metrics	30
3.4. Discussion	33
3.5. Conclusion	34
CHAPTER 4: STUDY B- AN IMPROVED TRANSFORMER-BASED MODEL FOR DETECTING PHISHING, SPAM AND HAM EMAILS: A LARGE LANGUAGE MODEL APPROACH.....	35
4.1. Introduction.....	35
4.2. Methodology	36
4.2.1. Data Collection and Preparation	38
4.2.2. Data Splitting	39
4.2.3. DistilBERT	40
4.2.4. RoBERTA	40
4.2.5. Improving the Training Process	41
4.2.6. Model Optimization	41
4.2.7. Learning Rate.....	43
4.2.8. Fine Tuning	44
4.3. Results.....	45
4.3.1. Evaluation metrics.....	46
4.3.2. Imbalanced Dataset Results	47
4.3.3. Balanced Dataset Results	50
4.3.4. Avoiding Overfitting	52
4.4. Discussion	53
4.5. Conclusion	54
CHAPTER 5: STUDY C- PERCEPTION AND EVALUATION OF TEXT-TO-IMAGE GENERATIVE AI MODELS: A COMPARATIVE STUDY OF DALL-E, GOOGLE IMAGEN, GROK, AND STABLE DIFFUSION.....	56
5.1. Introduction.....	56
5.2. Methods.....	57
5.2.1. Text to image diffusion models	58
5.2.2. DALL-E	59
5.2.3. Google Imagen.....	59

5.2.4. Stable Diffusion	60
5.2.5. GROK AI	60
5.3. Experiment And Result Analysis	60
5.4. Evaluation	61
5.4.1. Method A: Mathematical Evaluation	61
5.4.2. Method B: Human Evaluation	64
5.5. Discussion	70
5.6. Conclusion	71
CHAPTER 6: CONCLUSION.....	72
REFERENCES	74

LIST OF FIGURES

Figure 3. 1 ML service environment of cloud platform	20
Figure 3. 2 Flow Diagram of Overall Project	22
Figure 3. 3 A snapshot of the insurance dataset	23
Figure 3. 4 A snapshot of iris dataset	23
Figure 3. 5 A snapshot of real state home price prediction dataset	24
Figure 3. 6 A snapshot of wine quality red dataset	24
Figure 3. 7 A snapshot of wine quality white dataset.....	24
Figure 3. 8 Linear Regression Graph	25
Figure 3. 9 Amazon AWS Sage Maker Interface	26
Figure 3. 10 Code Snippet from AWS Sage Maker Jupyter Notebook.....	27
Figure 3. 11 Azure ML Studio Interface	27
Figure 3. 12 Azure ML Studio: Linear Regression Model Pipeline for Wine Quality Dataset	28
Figure 3. 13 Google Big Query Model Evaluation for Wine Quality Dataset	29
Figure 3. 14 Comparison Graph of R-Squared Value	30
Figure 4. 1 Overall Methodology	37
Figure 4. 2 Feature Distribution.....	39
Figure 4. 3 A Snapshot of Dataset Overview	39
Figure 4. 4 Basic Architecture of DistilBERT and RoBERTA.....	40
Figure 4. 5 Model Optimization.....	42
Figure 4. 6 Learning rate.....	44
Figure 4. 7 Code snippet of RoBERTA model DataLoader	45
Figure 4. 8 Fine-tuning process flow	45
Figure 4. 9 Comparison graph of baseline DistilBERT vs IPSDM performance (imbalanced dataset)	48
Figure 4. 10 Comparison graph of baseline RoBERTA vs IPSDM performance (imbalanced dataset)	49
Figure 4. 11 Comparison graph of Baseline DistilBERT vs IPSDM performance (balanced dataset)	51
Figure 4. 12 Comparison graph of Baseline RoBERTA vs IPSDM performance (balanced dataset).....	52
Figure 5. 1 Overall Method Flow Diagram	58
Figure 5. 2 Input Image of a Cat Sswimming	61
Figure 5. 3 Code snippet of FID Score Calculation.....	63
Figure 5. 4 Comparison of Mathematical Metrics	64
Figure 5. 5 Survey Questions Snapshot	66

Figure 5. 6 Groups with Significant Difference.....	69
Figure 5. 7 Groups with No Significant Difference	70

LIST OF TABLES

Table 3. 1 R squared value comparison	30
Table 3. 2 Error metrics of Microsoft Azure Platform	31
Table 3. 3 Error metrics of Google Cloud Platform.....	32
Table 3. 4 Average Error Metrics of Azure and GCP.....	33
Table 4. 1 Baseline DistilBERT vs IPSDM performance (imbalanced dataset)	47
Table 4. 2 Baseline RoBERTA vs IPSDM performance (imbalanced dataset)	49
Table 4. 3 Baseline DistilBERT vs IPSDM performance (balanced dataset)	50
Table 4. 4 Baseline ROBERTA vs IPSDM performance (balanced dataset)	51
Table 4. 5 Validation vs test accuracy	52
Table 4. 6 Validation vs test accuracy graph	53
Table 5. 1 Mathematical Evaluation Metrics	64
Table 5. 2 Group Summary	66
Table 5. 3 Group Differences	67
Table 5. 4 Tukey HSD Result.....	68

CHAPTER 1: INTRODUCTION

Machine learning (ML) and artificial intelligence (AI) algorithms are pivotal in predictive modeling and classification tasks while analyzing complex datasets and extracting valuable insights. These algorithms have revolutionized various domains by enabling accurate predictions, pattern recognition, and decision-making processes. In the domain of predictive modeling, ML and AI algorithms facilitate the development of predictive models that can forecast future trends, outcomes, or behaviors based on historical data. Similarly, in classification tasks, these algorithms categorize data into distinct classes or categories, empowering applications ranging from email filtering to medical diagnosis. Moreover, the advent of text-to-image generation has further expanded the horizons of AI applications, allowing for the creation of visual content directly from textual descriptions. This capability holds immense potential in diverse fields such as design, art, entertainment, and visual communication, where AI-driven image synthesis enhances creativity and productivity. ML and AI algorithms continue to drive innovation and transformation across industries, shaping the future of predictive modeling and image synthesis.

The first study on the regression-type machine learning modeling focuses on the performance evaluation of ML models of different cloud platforms. The significance of adopting cloud technology in enterprises is accelerating and becoming ubiquitous in business and industry. Due to migrating the on-premises servers and services into the cloud, companies can leverage several advantages such as cost optimization, high performance, and flexible system maintenance, to name a few. As the data volume, variety, veracity, and velocity are rising tremendously, adopting machine learning (ML) solutions in the cloud platform bring benefits from ML model building through model evaluation more efficiently and accurately. This study will provide a comparative performance analysis of the three big cloud vendors: Amazon Web Service (AWS), Microsoft Azure and Google Cloud Platform (GCP) by building regression models in each of the platforms. For validation purposes, i.e., training and testing the models, five different standard datasets from the UCI machine learning repository have been employed. This work utilizes the ML services of AWS Sagemaker, Azure ML Studio, and Google Big Query for conducting the experiments. Model evaluation criteria include measuring R-squared values for each platform, calculating the error metrics (Mean Squared Error, Mean Absolute Error, Root Mean Squared Error etc.) and comparing the results to determine the best-performing cloud provider in terms of ML service. The study concludes by presenting a comparative taxonomy of regression models across the three platforms.

The second study is on customizing and fine-tuning of large language models on a classification task which is identification of spam, ham, and phishing types of emails. Phishing and spam detection are long-standing

challenges that have been the subject of much academic research. Large Language Models (LLM) have vast potential to transform society and provide new and innovative approaches to solve well-established challenges. Phishing and spam have caused financial hardships and lost time and resources to email users all over the world. They frequently serve as entry points for ransomware threat actors. While detection approaches exist, especially heuristic-based approaches, LLMs offer the potential to venture into a new unexplored area for understanding and solving this challenge. LLMs have rapidly altered the landscape from business, consumers, and throughout academia and demonstrate transformational potential for society's potential. Based on this, applying these new and innovative approaches to email detection is a rational next step in academic research. In this work, we present IPSDM, an improved phishing spam detection model based on fine-tuning the BERT family of models to detect phishing and spam email specifically. We demonstrate our fine-tuned version, IPSDM, is able to better classify emails in both unbalanced and balanced datasets.

Finally, the third study is on text-to-image generation diffusion models, where we worked on the perception and evaluation of text-to-image generative AI models. Generative Artificial Intelligence (AI) model is a revolutionary type of AI capable of producing high-quality images based on textual inputs. These models utilize natural language processing (NLP) techniques and computer vision to understand and interpret the textual descriptions and then generate images that align with the given descriptions. This study evaluates four prominent text-to-image generative models- DALL-E, Google Imagen, Stable Diffusion, and GROK AI emphasizing on the text-to-image diffusion models. Using a comprehensive evaluation approach, we employ three mathematical formulas the Fréchet Inception Distance (FID), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) to assess image quality and realism across datasets collected from these AI platforms. Additionally, human evaluations are conducted to compare the perceptual impact of AI-generated images with mathematical metrics. Our findings contribute to the advancement of text-to-image synthesis and advocate for responsible AI development.

These three studies hold significant importance in the realm of machine learning and AI research, particularly in their focus on predictive and classification solutions deployed on cloud platforms. Firstly, the study on regression-type machine learning modeling contributes to our understanding of the performance evaluation of machine learning models across different cloud platforms. This insight is crucial for organizations and researchers seeking to optimize their model deployment strategies and choose the most suitable cloud infrastructure for their specific needs. Secondly, the research on customizing and fine-tuning large language models for classification tasks, such as identifying spam, ham, and phishing emails, addresses a pressing cybersecurity concern. Email classification plays a vital role in mitigating the risks associated with malicious activities and enhancing the accuracy and efficiency of these classification

models can significantly strengthen email security measures. Lastly, the investigation into text-to-image generation diffusion models explores the burgeoning field of AI-driven visual content generation. As the demand for AI-generated imagery continues to rise across various industries, understanding the perception and evaluation of text-to-image generative AI models becomes paramount. This research sheds light on the capabilities and limitations of such models, informing future advancements and applications in this rapidly evolving domain. Collectively, these studies contribute to advancing machine learning and AI technologies across various contexts, tackling real-world challenges and fostering innovation.

CHAPTER 2: LITERATURE REVIEW

2.1 STUDY A – LITERATURE REVIEW: REGRESSION MODELS EVALUATION ON CLOUD PLATFORMS: AWS VS AZURE VS GCP

There are many scholarly works on ML algorithms' comparisons, specifically on supervised models, but a few of them are noted in section 2.1.1. Later, the papers on the performance analysis of ML models on several open-source data mining tools are summarized in section 2.1.2. Furthermore, studies which focus on comparing different cloud vendors are summarized in section 2.1.3.

2.1.1. Supervised ML Algorithms' Performance Comparison

Abdulqader, et al. [1] presented several techniques of supervised machine learning algorithms for gene selection dataset. Various supervised algorithms: Support Vector Machine (SVM), Neural Network, K-nearest Neighbor, Naïve Bayes, Random Forest are elaborately discussed here. A survey has been conducted using supervised machine learning algorithms on gene selection methods. The performance has been measured as highest while using the SVM technique on four sets of microarray data. The lowest accuracy seems to be from Naïve Bayes which is 74.83% [1]. Similarly, from the experimental results of Meyer, et al. [2] on four machine learning algorithm (Random Forest, Neural Network, Averaged Neural Network and Support Vector Machine) on MSG SEVIRI data over Germany, SVM has relatively high error (123%) and its prediction rate is lower than other three. The R-squared values of each model increases significantly with the aggregation on to 24 hours [2].

Likewise, another comparative study was conducted by Osisanwo, et al. [3] where the classification indicates that SVM has the highest correctly classified instance (77.3021%). In another work, Maulud and Abdulazeez [4] described linear regression models elaborately and reviewed 23 papers on different types of linear regression: Simple, Polynomial and Multivariate Linear Regression (MLR) models. The paper discussed all the related equations for each model, and how the least square method is utilized to find the best fit line or curve. The higher accuracy is 99.89%, obtained by using MLRM technique on the Aero-Material dataset. The lowest accuracy (82.15) is found for the Pima Indian Diabetes dataset while using the same MLRM method [4]. Moreover, on the types, techniques and implementations of machine learning algorithms, Wang, et al. [5] discussed in detail on linear regression models. It is noted that principal components analysis (PCA) has significance in reducing data dimensionality of unsupervised learning. Other important concepts like local representation, interpolation with kernel and smoothness prior also have impacts on predictive functions which are clearly demonstrated in this paper [5].

Again, Kolisetty and Rajput [6] aimed to facilitate the understanding of ML's significance for large data analysis. High volume data processing needs more computational power and increased hardware, it also becomes high in cost. Impact of these perspectives on real time data analysis is a major factor to be considered. This paper also suggests the opportunities from the encouraging features development in the field of machine learning with the use of big data [6]. On the other hand, Asim, et al. [7] have adopted the three different machine learning approaches: lazy learning, decision tree with different variants and ensembling technique identifying professional bloggers. While working with D tree classifiers, famous algorithms for D trees, such as Random Tree, REP Tree, Random Forest, Simple Cart, NB Tree, and AD Tree, are used. Evaluating the results, it is found that the best correctly classifiers are Random Tree and Random Forest with 92% accuracy rate having only 8% error [7]. Moreover, Love [8] focused on the modes of unsupervised learning model: intentional and incidental, and their relationships with supervised classification learning. After running three algorithms on the collected data, four types were observed, with type 2 supervised learning having the highest accuracy (95%). The overall work after result evaluation summarizes that there are no advantages of engaging intentional unsupervised learning over incidental except the target concept is low in dimension and non-linear [8].

2.1.2. ML Model Evaluation on Open-source Data Mining Tools

Numerous research works have compared the performances of machine learning algorithms on open-source data mining tools. Like, Ratra and Gulia [9] 2020 aimed at analyzing data mining tools, Orange, and WEKA by implementing three classification algorithms: Naïve Bayes, K-nearest Neighbor and Random Forest. While comparing the precision metrics, the results show that WEKA has a higher percentage than Orange. Naïve Bayes performs the highest in both platforms which is 83.7% in WEKA and 82.4% in Orange [9]. Similarly, Kodati and Vivekanandam [10] presented a comparative review on WEKA and Orange tools for mining and analyzing of Heart Disease Dataset from UCI data repository. Authors have conducted an analytical study on four machine learning algorithms: Naïve Bayes, SMO, Random Forest, and K-Nearest Neighbor. When the dimension of the inputs is high, Naïve Bayes has the highest performance in Precision and Recall in WEKA and Orange tools. The Precision and Recall values of K-Nearest Neighbor is 0.753 and 0.752 in WEKA, whereas in Orange it is consecutively 0.58 and 0.547. K-Nearest Neighbor has the lowest performance among the four [10]. Moreover, Jamal, et al. [11] developed boosting methods on WEKA and Orange tools to predict heart disease death rate with an accuracy of 72% in Orange tool and 77% in WEKA. In another study, Kavitha, et al. [12] worked on regression models and compared two potential functions for linear regression algorithm: SMOReg function and LeastMedSq function. Open University Learning Analytics dataset has been used which is multivariate, time series and sequential. While comparing both models, it is obvious that the SMO regression function took 2.42 seconds which is less than

the LeastMedSq function of linear regression (3.29 seconds). However, all the error metrics (mean absolute error, relative absolute error, root mean squared error) are less in LeastMedSq linear regression than SMO regression. The result concludes that the LeastMedSq function performs better for linear regression algorithm [12].

Moreover, another experiment was conducted by Rajagopal, et al. [13] on UNSW NB-15 dataset for intrusion detection. The experiment is conducted in Azure Machine Learning Studio to evaluate the models where 10-fold cross-validation technique has been applied. Four classification models (Random Forest, Decision Tree, Naïve Bayes and SVM) have been compared while Apache Spark is used as a processing paradigm. Result indicates that as a classifier, Decision Forest performs the highest. Moreover, the eight two class models took minimal time for training which ranged in 6 to 9 seconds. The study also emphasized that Microsoft Azure Machine Learning Studio (MAMLS) can be considered a potential integrated development environment for handling large volumes of datasets [13].

2.1.3. Cloud Vendors' Comparison: Services and Design Taxonomies

Related to cloud platform comparison based on general services, Kaushik, et al. [14] discussed and compared among three large cloud vendors: Amazon AWS, Microsoft Azure and Google Cloud Platform. It elaborated the different computing platforms of cloud as can be segmented into two elements front end and back end. Authors have tabulated all the prices of services that are provided by these three cloud vendors where it is obvious that Azure is the most expensive for general purpose instances. However, AWS has the cheapest options for choosing instances. For testing the performance, Phoronix Test Suite3 was adopted on Linux systems. The test processes were completed in Apache, RAM speed and Dbench benchmark. In the Apache measurement, it is seen that Azure handles more HTTP requests better than the two. Again, in the Dbench test, AWS and Azure differences are negligible while GCP performs less than the two [14]. Similarly, in another survey by Alkhatib, et al. [15], the finding shows that market shares of AWS (32%) is larger than Azure (19%) and GCP (7%). In terms of security, AWS has AWS Security Hub, Azure uses Azure Security Center and Google has their Cloud Security Command Center. While considering the weakness, Azure seems most expensive which can cut down their customers, AWS is sometimes considered as difficult to use and GCP has comparatively fewer features than others [15].

Another taxonomy of services is provided by Sikeridis, et al. [16] on the four dominants in perspective of market share and the sub-services designating storage, data pipeline, analytics, databases, machine learning etc. While using cloud services, customers can choose to pay per usage model for billing. Major cloud vendors provide a combination of low cost (Zero installation and maintenance cost) and high performance. Based on the service types of computer services and virtual machines, the offered services by

Amazon, Microsoft, Google, and IBM have been tabulated where it is found that Max memory is used by Amazon (1952 GB: X1) and the lowest is IBM (242 GB). The virtualization of hypervisor-based, and container based have been drawn as well. The serverless computing services for each provider are Amazon: AWS Lambda, Microsoft: Azure Function, Google: Cloud Functions and IBM: Open Whisk. All the four providers have no SQL, petabyte-scale and relational databases [16].

So far, other studies focused on the comparisons of ML algorithms in open-source data mining tools or offline tools. However, the performance evaluation of these models in cloud platforms is yet to be measured. Hence, our first study aims to build such an analysis among Azure, AWS, and GCP by conducting experiments on individual platforms.

2.2 STUDY B – LITERATURE REVIEW: An Improved Transformer-based Model for Detecting Phishing, Spam and Ham Emails: A Large Language Model Approach

2.2.1. Machine Learning and Deep Learning-based Methods

Numerous machine learning and deep learning-based spam email detection and classification applications have been carried out over the past few decades by many researchers. In such studies [17-22], authors have proposed, reviewed, and evaluated spam filtering models where the classification models are based on traditional machine learning algorithms, i.e., Naïve Bayes, Random Forest, SMV mostly. Govil, et al. [17] created a dictionary named “stopwords” to remove the helping verbs from email. Then, the algorithm is executed for checking the possibility of being spam or not. A machine learning classifier, Naïve Bayes has been applied for the identification purpose where non-spam emails were classified as spam, 1 and non-spam, 0 [17].

Similarly, Chen, et al. [18] have evaluated machine learning algorithms for detecting spam tweets. A large dataset containing around 600 million public tweets have been collected first. Later, Trend Micro’s Web Reputation System was applied to label the spam emails. Experiments on different data sizes revealed that TP rate is increased from 78% to 85% following KNN and 70% to 75% following the Random Forest classifier. Another potential finding is the classifier could detect continuously sampled spam tweets better than randomly selected tweets [18]. In the similar context of Twitter spam detection, Wu, et al. [23] introduced a WordVector Training-based model with a classification accuracy of around 80%. This work has achieved an average 30% higher F-measures compared to other existing models.

Moreover, Guzella and Caminhas [21] reviewed the textual and image-based spam email filtering approaches focusing on designing new filters. Most common method selecting the feature is information gain and this way of collecting features might increase accuracy. Regarding datasets, SpamAssassian and LingSpam are considered the most popular ones, whereas TREC corpora can produce a more realistic online

setting. Moreover, Chetty, et al. [24] proposed a deep learning-based model combining Word Embedding and Neural Network aiming to detect spams from various text documents. Naïve Bayes model is considered as the baseline model for comparing with the deep learning model. Datasets were collected from UCI machine learning repository for developing the models. For the SMS dataset, the highest performance (accuracy 98.7%) is achieved from the combined Word Embedding and neural network model. Apart from the supervised learning approaches, there are numerous works on unsupervised modeling as well [25-30]. Utilizing Modified Density-Based Spatial Clustering of Applications with Noise (M-DBSCAN), 97.848% accuracy has been obtained by Manaa, et al. [26]. An online unsupervised spam detection scheme, SpamCampaignAssassin (SCA) could detect around 92.4% spam for DEPT trace email dataset [25].

2.2.2 Transformer Model-based Approaches

The research works and literature landscape on transformer-based methods are relatively limited. The domain of fine-tuned transformers or attention mechanism techniques for identifying spam emails is still an emerging new field. Related to this specific area, Yaseen [31] has introduced an effective word embedding technique for spam classification. Pre-trained transformer, BERT is fine-tuned to detect the spam emails from non-spam emails. Deep Neural Network with BiLSTM is considered as a baseline model to compare the model. Two open-source datasets from UCI machine learning repository and Kaggle have been employed to train and test the model. The proposed model could achieve 98.67% classification accuracy. Similarly, Liu, et al. [32] have developed and evaluated a modified spam detector transformer using the publicly available datasets, Spam Collection v.1 and UtkMI's Twitter Spam Detection Competition dataset. This model could obtain 98.92% accuracy with a recall and F1 scores rate respectively, 0.9451 and 0.9613.

Furthermore, Guo, et al. [33] and Tida and Hsu [34] focused on BERT models implying the significance of self-attention mechanism. Guo, et al. [33] utilized two public datasets, Enron [35] and a simple spam email classifier dataset from Kaggle for classifying ham or spam emails using pre-trained BERT model. Similarly, an Universal Spam Detection Model (USDm) has been developed and tested using four publicly available datasets: Ling-spam dataset [36], spam text dataset from Kaggle, Enron dataset and spam assassin dataset. This model has gained overall accuracy of 97% with 0.96 F1 score [34]. Moreover, for detecting phishing URL and cyberbullying identification models, researchers worked on fine-tuning BERT-based models [37-40]. Wang, et al. [37] have scrapped 2.19 million pieces of URL data from PhishTank while pre-training PhishBERT model. This model exhibited 92% accuracy in detecting phishing URLs. Similarly, Maneriker, et al. [38] fine-tuned BERT and RoBERTa models and proposed a URLTran transformer. Microsoft Edge and Internet Explorer browsing telemetry data have been employed for training, testing, and validating purpose. Down sampling method is applied for balancing the datasets where the final training dataset had

77,870 URLs. The final models had a True Positive Rate (TPR), 86.80% compared to the baseline models URL-Net [41] and Texception [42].

In the current state of machine learning, deep learning, and transformer-based models, a notable gap persists in the literature where the fine-tuning of BERT families has not been extensively explored. Despite the remarkable achievements of BERT-based architectures in several natural language processing tasks, there remains a lack of comprehensive research on fine-tuning these models to address specific domain challenges. In our second work, we aim to close this gap by implementing fine-tuning techniques on BERT variants such as DistilBERT and RoBERTA. By customizing and fine-tuning these models on datasets relevant to specific applications, we seek to enhance their adaptability and effectiveness in addressing sophisticated real-world problems in complex domains, thereby contributing to the advancement of machine learning and deep learning methodologies.

2.3 STUDY C – LITERATURE REVIEW: PERCEPTION AND EVALUATION OF TEXT-TO-IMAGE GENERATIVE AI MODELS: A COMPARATIVE STUDY OF DALL-E, GOOGLE IMAGEN, GROK, AND STABLE DIFFUSION

The research landscape in text-to-image generation is still relatively nascent, with a limited number of works exploring this emerging field. While interest in text-to-image synthesis has been growing, particularly in recent years, the volume of literature remains relatively modest compared to more established areas of machine learning and computer vision. In the context of image tuning, et al. [43] introduced UniTune, an innovative approach to image editing that accepts any image along with a textual description of the desired edit. It is capable of executing the modification while preserving the original image's quality. Unlike other methods, UniTune doesn't rely on additional inputs like masks or sketches and can handle multiple edits without retraining. In evaluations against another similar potential model, SDEdit, UniTune demonstrated a clear advantage, with a 72% preference over SDEdit's 28%. These results indicate that while both methods excel when edits are minor, UniTune outperforms significantly in scenarios requiring substantial pixel alterations, such as object duplication, movement, or resizing. In another study related to versatile diffusion model, Xu, et al. [43] extended the original single-flow diffusion pipeline into a versatile multi-task multimodal network called Versatile Diffusion (VD), capable of handling various tasks such as text-to-image and image-to-text conversions within a single unified model. VD comprises three key components: a diffuser that operates within a multi-flow multimodal framework, variational autoencoders (VAEs) for converting data samples into latent representations, and context encoders for embedding contextual information. In comparison to existing models like CogView, LAFITE, GLIDE, and Make-a-Scene, VD demonstrates superior FID performance, with a score of 11.21 ± 0.03 compared to 11.10 ± 0.09 .

Furthermore, another method has been developed by Elarabawy, et al. [44] for featuring multiple easily adjustable hyperparameters, enabling a diverse array of real image edits. This method, termed as optimization-free and zero fine-tuning, relies on text-based semantic instructions for flexible editing. Unlike approaches generating numerous outputs and relying on additional mechanisms for filtering, this method allows for systematic modulation of target edits. Moreover, Ruiz, et al. [45] introduced a method called Dream Booth, which synthesizes new renditions of a subject using a small set of subject images and a text prompt as guidance. This framework is developed by fine-tuning a text-to-image model with the input images and text prompts containing a unique identifier followed by the class name of the subject. Two metrics, CLIP-I and DINO were used to evaluate the performance. Dream Booth (based on the Imagen model) achieves higher scores for both subject and prompt fidelity compared to Dream Booth (based on Stable Diffusion), nearing the upper limit of subject fidelity achievable with real images.

For image synthesis, several research have been carried on blended diffusion models which combine ideas from diffusion models and autoregressive models to achieve high-quality image generation [46, 47]. A latent diffusion model (LDM) was designed by Avrahami et al. [48] to accelerate the diffusion process by functioning within a lower-dimensional latent space. This design eliminates the necessity for resource-intensive CLIP gradient computations at each diffusion step, thereby enhancing efficiency without compromising on the quality of image synthesis. Numerous researchers have directed their attention toward transformer-based applications to achieve rapid and high-resolution image synthesis[48-50]. Ding, et al. [51] designed a solution leveraging hierarchical transformers and local parallel autoregressive generation. They pretrained a 6-billion-parameter transformer using a straightforward and adaptable self-supervised task, namely a cross-modal general language model called CogLM, and further refined it for swift super-resolution tasks. Their novel text-to-image system, CogView2, exhibits highly competitive generation capabilities when compared to contemporary state-of-the-art models like DALL-E-2, while also inherently supporting interactive text-guided editing of images.

Another two-stage model has been developed by Ramesh, et al. [52] where the first stage generates a CLIP image embedding from a provided text caption, and the second stage consists of a decoder that generates an image conditioned on this embedding. Moreover, Latent diffusion models offer significant advantages for tasks such as image inpainting, class-conditional image synthesis, as well as competitive performance across various other tasks like unconditional image generation, text-to-image synthesis, and super-resolution. In one such study, Rombach, et al. [53] analyzed the behavior of their latent diffusion models (LDMs) across different down sampling factors. These models are denoted as LDM-f, where LDM-1 corresponds to pixel-based diffusion models. This LDM (Latent Diffusion Model) has achieved new state-of-the-art scores for tasks such as image inpainting and class-conditional image synthesis. Moreover, it

demonstrates highly competitive performance across a range of tasks, including unconditional image generation, text-to-image synthesis, and super-resolution.

Chapter 3:

Study A- Regression Models Evaluation on Cloud Platforms: AWS vs Azure vs GCP

3.1. INTRODUCTION

In the technological revolution, cloud computing has been the biggest buzz, impacting business and industry from every conceivable angle. The tendency to migrate to the cloud from local data centers, also known as on-premise [54] is quite a common scenario in almost every institution. Cloud computing plays a significant role in managing complex and sophisticated IT infrastructure, scaling up and down required resources, and focusing more on business operations. For working in the cloud platforms, migrating all on-premises data or applications to cloud, managing the virtualized resources efficiently [55, 56], and optimizing the cloud computing, it is an intelligent way to work with one of the cloud partners. In terms of big data and increased workloads, integrating machine learning solutions to the cloud and to deploy in the enterprise applications have added many advantages including flexibility, cost effectiveness and efficiency. To build, train, deploy, and test ML models with low coding experience, less maintenance, and required expertise, companies increasingly rely on cloud vendors [57]. Figure 3.1 represents the general ML service structure of cloud environment. Larger cloud computing service providers offer multiple options to implement the intelligent features in the enterprise applications which don't demand highly skilled professionals to work with AI or ML projects thereby offering cost savings.

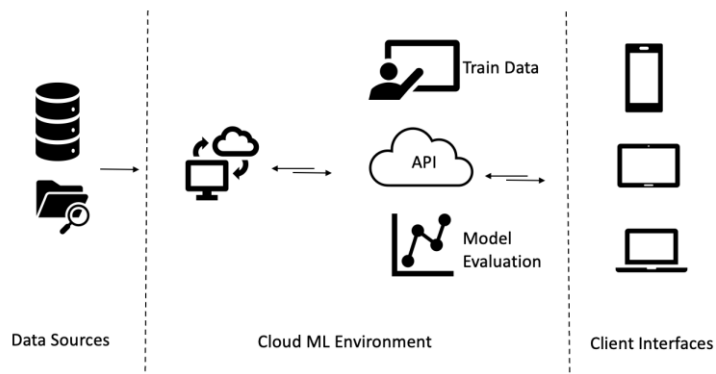


Figure 3. 1 ML service environment of cloud platform

Enabling a computer system or machines to learn without explicit intervention or instruction of humans can be defined as Machine learning. Learning by the machine itself using algorithms or statistical approaches or models for analyzing data patterns and predicting outcomes is the basic idea behind this. ML

and artificial intelligence are making a steady approach into every industry sector such as enterprise, health care, education, engineering, and manufacturing [58-60]. These technologies are becoming integral in language processing and natural language processing (NLP) tasks. As organizations seek to leverage data-driven insights and automation to enhance efficiency, decision-making, and innovation, ML and AI are playing an increasingly vital role in transforming various aspects of business operations and service delivery across diverse domains [61].

Deployment of machine learning models in the cloud brings extended benefits by removing many technical hurdles. Algorithms are widely adopted for handling resource management, scheduling tasks, and optimizing energy ML [62]. Cloud providers like Azure, AWS, and GCP offer to work with a variety of machine learning algorithms, still there are significant differences in terms of front-end interface, background setup etc. AWS is called the most mature provider with a range of offers for small development companies, large enterprises and even governments. They have the largest set of services. Microsoft Azure is popular for their drag and drop interface, which doesn't require prior coding experience. The geographic coverage of Azure appears broader than others. GCP on the other hand is the smallest of the big three providers, however, it provides a robust set of solutions to any kind of application.

Since our data is increasing in an enormous volume, the significance and dependency on machine learning is also escalating gradually. The greater the data volume, the more computational power is required, and machine learning is playing a vital role in this case. For machine learning workloads, cloud is providing a pay-per-use model which is very cost effective. Certain barriers exist to bringing machine learning solutions to enterprises. To build, train and deploy models specialized skills are required. Apart from this, there is a high demand of computational power and special hardware adding up to high cost for development, labor, and infrastructure. All these barriers can be overcome with cloud computing solutions. Companies can leverage the highest speed and Graphical Processing Unit (GPU) power while training and experimenting machine learning models in cloud environment. Similarly, obstacles to storing high volume data have also been overcome. Not only the large enterprises, but also small to medium companies are taking the advantages of this cloud computing technology [63]. While choosing cloud vendors for machine learning service, reviewing the performance and offerings for each platform is recommended. During the collaboration with ML solutions, different cloud providers use different backend services, frameworks, and algorithms. Hence, performance of ML models might vary from platform to platform. It is strongly advised to understand and analyze the features and performance of the different cloud vendors while choosing one for machine learning implementation.

This study addresses the performance evaluation of ML models on different cloud platforms. At first, the relevant literatures to our work are summarized. Next, our method collected five standard datasets from the

UCI machine learning data repository, examined the datasets and prepared data to input as for feeding into linear regression models, segregated the data for training and testing purposes, developed linear regression models in the three cloud platforms (Azure, AWS and GCP). Following the illustration of our methods, the results evaluate the models using the metrics like R Squared Value, Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) on the three largest cloud platforms. Our research will assist enterprises in effective decision-making when choosing a suitable cloud vendor when they consider leveraging cloud technology infused with machine learning solutions.

3.2. METHODS

The overall flow diagram of the project has been presented in figure 3.2, where in the beginning of the process, a total of five datasets have been collected from UCI Machine Learning data repository. All these data are standard and publicly available. The WEKA tool has prepared data to feed as inputs of the machine learning models. It has been ensured that there are no null or missing values in our data samples. After analyzing the data, linear regression models were built consecutively. At first ML model was built in Amazon AWS using the AWS Sage maker service. Then utilizing Azure ML studio, all the pipelines for ML models have been created. Later, taking the service of Google Big Query, model creation was performed. Finally, all the models have been evaluated by calculating the metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Coefficient of determination or R-square etc.

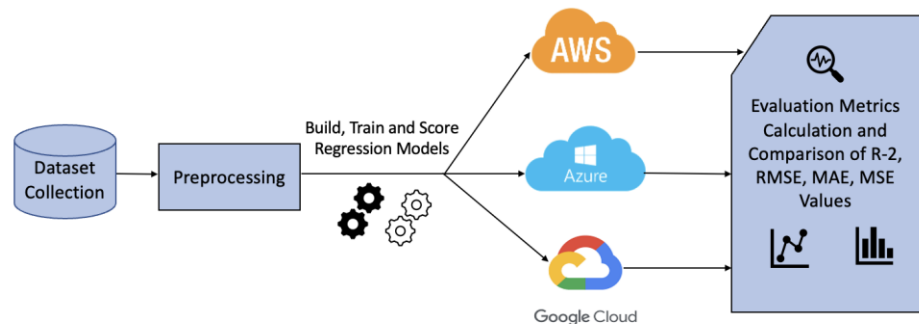


Figure. 3.2.

Figure 3. 2 Flow Diagram of Overall Project

3.2.1. Datasets

All the datasets have been collected from UCI Machine Learning Data Repository, standard for regression model analysis. UCI machine learning repository has a large collection of datasets, data generators and domain theories which scholars, students, and researchers widely used as an authentic source of datasets.

Insurance Dataset. There are a total of 6 features (age, sex, bmi, children, smoker, and region) and one target variable, ‘charges’ in the data set. It has 1437 tuples in total. As sex, smoker and region attributes are non-numeric, numeric values have been assigned to these variables to make all data samples numeric [20]. It is a widely used dataset available publicly for exploratory data analysis and hypothesis testing on specifically regression models.

	age	sex	bmi	children	smoker	region	charges
1							
2	19	1	27.9	0	1	4	16884.924
3	18	0	33.77	1	0	3	1725.5523
4	28	0	33	3	0	3	4449.462
5	33	0	22.705	0	0	2	21984.4706
6	32	0	28.88	0	0	2	3866.8552
7	31	1	25.74	0	0	3	3756.6216
8	46	1	33.44	1	0	3	8240.5896
9	37	1	27.74	3	0	2	7281.5056
10	37	0	29.83	2	0	1	6406.4107

Figure 3. 3 A snapshot of the insurance dataset

Iris Dataset. This dataset contains a total of 4 independent variables (sepal width, petal length, and species) and one dependent variable, ‘sepal length’. The dataset has a total 150 data samples. The species are of Iris-setosa, Iris-versicolor, and Iris-virginica. Based on these 4 features, the regression model will make the prediction on the sepal length variable [21].

	sepal width	petal length	petal width	species	sepal length
1					
2	3.5	1.4	0.2	Iris-setosa	5.1
3	3	1.4	0.2	Iris-setosa	4.9
4	3.2	1.3	0.2	Iris-setosa	4.7
5	3.1	1.5	0.2	Iris-setosa	4.6
6	3.6	1.4	0.2	Iris-setosa	5
7	3.9	1.7	0.4	Iris-setosa	5.4
8	3.4	1.4	0.3	Iris-setosa	4.6
9	3.4	1.5	0.2	Iris-setosa	5
10	2.9	1.4	0.2	Iris-setosa	4.4

Figure 3. 4 A snapshot of iris dataset

Real Estate Home Price Prediction Dataset. Real estate house price prediction is also a standard UCI machine learning data repository dataset. It has a total 6 feature variables and one target variable, which is ‘unit area price’. Depending on the 6 features, the model makes predictions of the house price on a unit area. All the values of this dataset are numerical [22].

	x1	x2	x3	x4	x5	x6	Unit Area Price
1							
2	2012.917	32	84.87882	10	24.98298	121.54024	37.9
3	2012.917	19.5	306.5947	9	24.98034	121.53951	42.2
4	2013.583	13.3	561.9845	5	24.98746	121.54391	47.3
5	2013.5	13.3	561.9845	5	24.98746	121.54391	54.8
6	2012.833	5	390.5684	5	24.97937	121.54245	43.1
7	2012.667	7.1	2175.03	3	24.96305	121.51254	32.1
8	2012.667	34.5	623.4731	7	24.97933	121.53642	40.3
9	2013.417	20.3	287.6025	6	24.98042	121.54228	46.7
10	2013.5	31.7	5512.038	1	24.95095	121.48458	18.8

Figure 3. 5 A snapshot of real state home price prediction dataset

Wine Quality-Red Dataset. It has a total of 1598 data rows with twelve independent variables and one dependent variable, quality. As the dataset has all numeric variables, it has been kept unmodified while loading for further model build-up [23].

	fixedAcidity	volatileAcidity	citricAcid	residualSugar	chlorides	freeSulfurDioxide	totalSulfurDioxide	density	pH	sulphates	alcohol	quality
1												
2	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
3	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
4	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
5	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
6	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
8	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5
9	7.3	0.65	0	1.2	0.065	15	21	0.9946	3.39	0.47	10	7
10	7.8	0.58	0.02	2	0.073	9	18	0.9968	3.36	0.57	9.5	7

Figure 3. 6 A snapshot of wine quality red dataset

Wine Quality-White Dataset. This is also a standard dataset for working with regression models. The dataset contains a total of 13 attributes. Quality is the target or dependent variable here, which is predicted using the 12 independent variables. The number of rows in the dataset is 4897. The values are all numeric, and there are not any missing values. Hence, this dataset has been used without any modification [24].

	fixedAcidity	volatileAcidity	citricAcid	residualSugar	chlorides	freeSulfurDioxide	totalSulfurDioxide	density	pH	sulphates	alcohol	quality
1												
2	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
3	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
4	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
5	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
6	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
7	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
8	6.2	0.32	0.16	7	0.045	30	136	0.9949	3.18	0.47	9.6	6
9	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
10	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
11	8.1	0.22	0.43	1.5	0.044	28	129	0.9938	3.22	0.45	11	6
12	8.1	0.27	0.41	1.45	0.033	11	63	0.9908	2.99	0.56	12	5

Figure 3. 7 A snapshot of wine quality white dataset

3.2.2. Cloud Platforms

With the evolution of computing and technology, cloud computing has been a blessing for implementing and working with machine learning algorithms. Different platforms have different orientations and interfaces for building models. For building and training cost-effective, memory-efficient solutions with simple or complex machine learning algorithms, several cloud platforms like Azure, AWS and GCP have their different service levels. In this work, we will implement regression models using above five different datasets for running the experiment into three platforms (Azure, AWS and GCP).

3.2.3. Algorithm

Linear regression is one of the most famous supervised algorithms used for predictive analysis. It makes predictions for real or continuous or numeric values like age, salary, product price etc. The formula stands as:

$$Y = a + bX$$

Where Y is dependent or target variable, X is predictor or independent variable, and b is the slope of the line. This algorithm provides a straight line between these variables X and Y, as shown in figure 3.8.

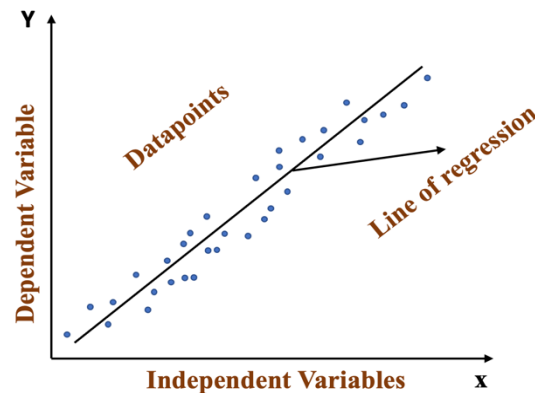


Figure 3. 8 Linear Regression Graph

We will be working with a linear regression model and analyzing the performance metrics in three cloud environments. This has been ensured that all the datasets were kept the same for all three platforms. Also, the procedures, data splitting process etc. were kept similar while building the models.

3.2.4. Procedure

Amazon AWS. In the AWS platform, the Sage maker service has been used which is a fully managed machine learning Amazon Elastic Compute Cloud (Amazon E2C) Compute Instance. With Sage Maker, different machine learning models can be easily and quickly built, trained, tested, and deployed in the production ready environment. A Jupyter notebook instance has been created to build linear regression models. Initially the datasets have been loaded into Amazon S3 bucket, which is the public cloud storage available in Amazon Web Service. The datasets have been split into 70%-30% ratio for training and testing using sk-learn functions. Using Sage maker boto3 services, the model was trained, and the model summary was evaluated. The interface of the AWS sage maker is presented in Figure 3.9.

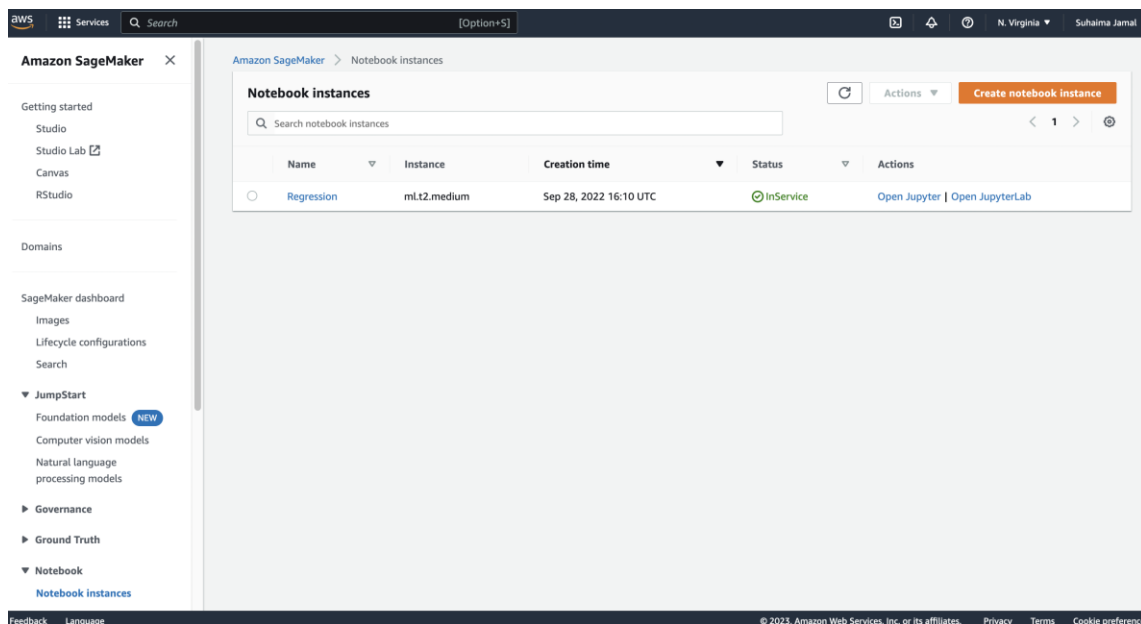


Figure 3. 9 Amazon AWS Sage Maker Interface

```
In [5]: from sklearn.model_selection import train_test_split
training_data = df.sample(frac=0.7, random_state=25)
testing_data = df.drop(training_data.index)

print(f"No. of training examples: {training_data.shape[0]}")
print(f"No. of testing examples: {testing_data.shape[0]}")

No. of training examples: 937
No. of testing examples: 401

In [6]: import boto3
import sagemaker
from sagemaker import get_execution_role

sagemaker_session = sagemaker.Session()
role = sagemaker.get_execution_role()

In [9]: import statsmodels.formula.api as smf

In [11]: model=smf.ols('charges ~ age + sex + bmi + children + smoker + region', data=
```

Figure 3. 10 Code Snippet from AWS Sage Maker Jupyter Notebook

Microsoft Azure. In our experiment, Azure Machine Learning Studio was used to create the machine learning pipelines in the designer and authoring section. Azure blob storage has been used to store the datasets. The default computing instances of Azure have been utilized here. Randomization and splitting data ratio were kept as 70%-30%. The Azure ML Studio interface and the flow of one of our pipeline creations are shown consecutively in figures 3.11 and 3.12.

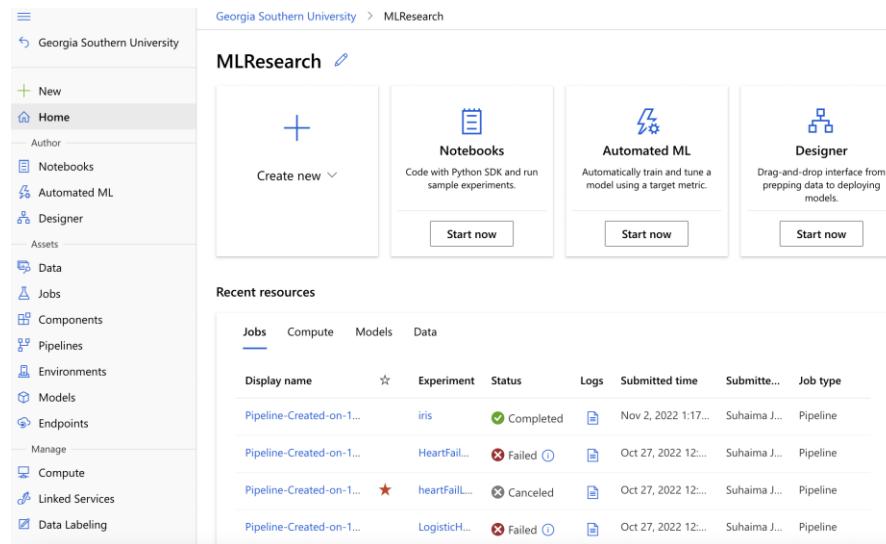


Figure 3. 11 Azure ML Studio Interface

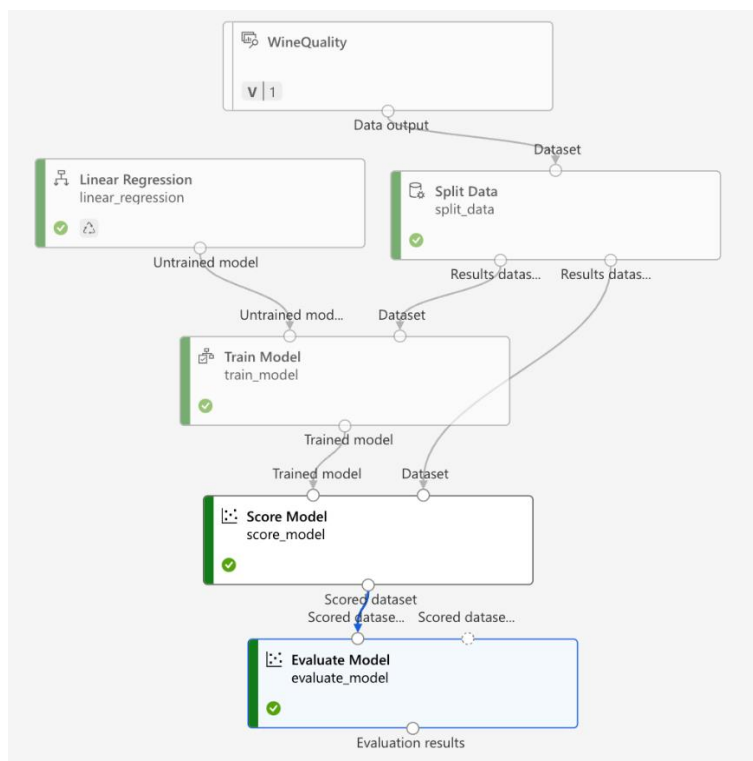


Figure 3. 12 Azure ML Studio: Linear Regression Model Pipeline for Wine Quality Dataset

Google Cloud Big Query. Using Google Big Query, our linear regression model is built in Google Cloud. Google Big Query is a serverless and cost-effective warehouse that works with big data and gets insights by extracting features. After building the model, all the results have been evaluated for our five datasets. The snapshot of the evaluation criteria using insurance dataset has been attached in figure 3.13 where different error metrics like Mean Absolute Error, Mean Squared Error, Mean Squared Log Error etc. can be found.

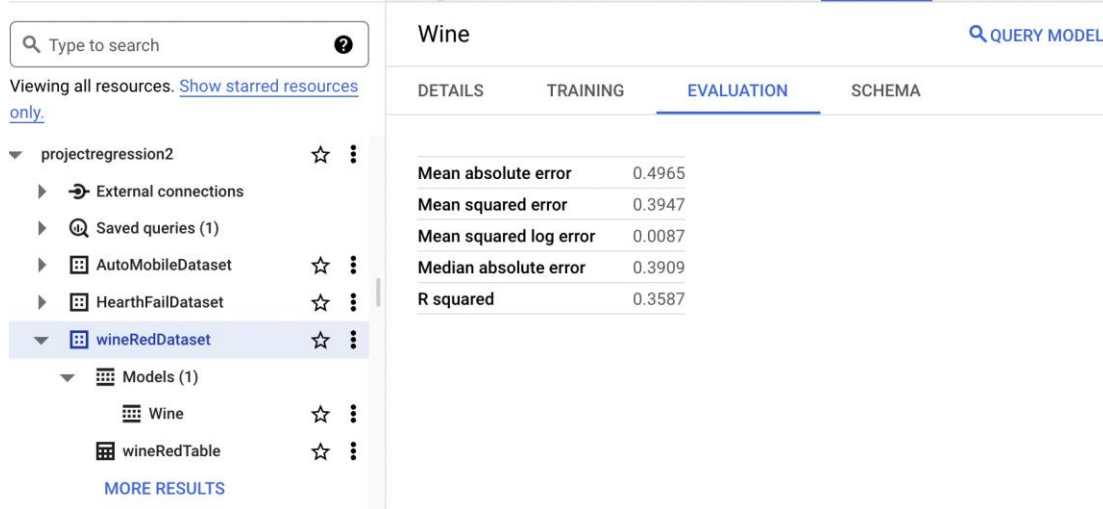


Figure 3. 13 Google Big Query Model Evaluation for Wine Quality Dataset

3.3. EXPERIMENTAL RESULTS

We have used five different datasets standard for regression models from UCI machine learning and run our experiments in three cloud platforms (Azure, AWS and GCP). First, linear regression models have been built using AWS Sage maker Jupyter Notebook instance. Then in Azure ML studio, the pipelines of the models have been created for all the five datasets and the evaluation results are recorded. All the results were collected from these two platforms. Later, we worked on Google Big Query for building and evaluating our model in Google Cloud Platform. Finally, the comparison of R squared values and different error metrics are calculated and compared among the three platforms' results.

3.3.1. R Squared Value or Coefficient of Determination

This statistical measure in the regression model determines the variance proportion of the dependent variable which the independent variable can explain. This value can determine the fitness of the model. The higher R-squared value is the better for the model fitness.

R-Squared Formula is defined in equation 3.1:

$$R_{\text{Squared}} = \frac{SS_{\text{Regression}}}{SS_{\text{Total}}}$$

$SS_{\text{Regression}}$ = Sum of squares due to regression

SS_{Total} = Total sum of squares

Interpretation of R-Squared values. The higher the R-squared value, the better the regression models fit with the testing data. When the values of x account for R-squared = 1, all the variations of y values.

When $R\text{-squared} = 0.5$, 50% of the variations of y values are accounted for by the values of x . When $R\text{-squared} = 0$, None of the variation of y is accounted for by x .

Table 3. 1 R squared value comparison

Platform	Dataset Name					Average
	Insurance	Iris	Real Estate Home Price	Wine Quality Red	Wine Quality White	
Azure	0.745	0.870	0.594	0.296	0.281	0.557
AWS	0.751	0.868	0.582	0.361	0.261	0.565
GCP	0.784	0.868	0.582	0.359	0.261	0.571

Our experiments obtained R -squared values from Azure, AWS and GCP, which have been tabulated in table 3.1. At first R -squared values are calculated for each of the dataset and then the average is calculated to compare the results. The higher average R -squared value is obtained from GCP (0.571) and then AWS (0.565). However, Azure (0.557) performs comparatively less than the other two. The bar chart in figure 3.14 shows the performance comparison among these three cloud providers.

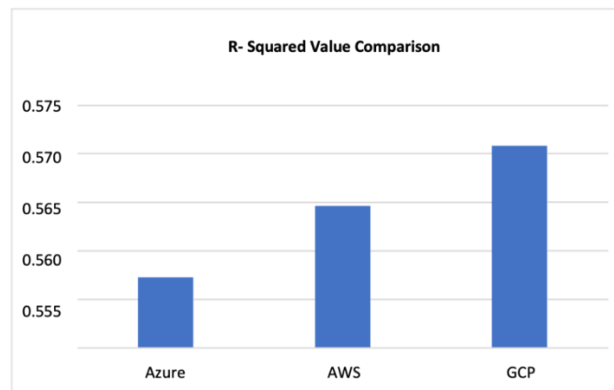


Figure 3. 14 Comparison Graph of R -Squared Value

3.3.2. Error Metrics

Error metrics are used to quantify performance of models and provide ways for forecasting to compare different models quantitatively. These metrics give a precise gauge on the performance of the models. There

are few common error metrics for reporting and evaluating linear regression model performance, these are: Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Squared Log Error (MSLE).

Root Mean Squared Error (RMSE). This calculates the square root of average squared distance, which is the difference between the actual and predicted values. This is a popular evaluating metric for regression models as it calculates how the prediction is close to the actual average and indicates the effects of large error. Large errors will always have a significant impact on the RMSE value. The formula for RMSE is as below equation 3.2:

$$RMSE = SD_y \sqrt{(1 - r^2)}$$

SD is the standard deviation. The lower the RMSE value, the better the model fits to the dataset.

Mean Absolute Error (MAE). Mean Absolute Error is the loss function of a regression model. The loss denotes the mean of the absolute differences between actual and predicted values or, the deviation from the actual value. Less MAE value is better and if it tends to zero the model is more accurate. MAE formula is as following equation 3.3:

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y - \underline{Y}|$$

Here Y is the output value.

\underline{Y} is the predicted value.

n is the total data points.

Mean Squared Log Error (MSLE). The measurement of the ratio between log-transformed actual and log transformed predicted value of a model can be noted by Mean Squared Log Error (MSLE) as shown in equation 3.4.

$$MSLE = \frac{1}{N} \sum_i^N (\log_e(1 + y_i) - \log_e(1 + \underline{y}_i))^2$$

Table 3. 2 Error metrics of Microsoft Azure Platform

Datasets

Error Metrics	Insurance	Iris	Real Estate Home Price	Wine Quality Red	Wine Quality White
Mean Absolute Error (MAE)	7197.14	0.213	6.273	0.554	0.596
Relative Absolute Error	0.4698	0.340	0.609	0.781	0.878
Relative Squared Error	0.255	0.130	0.406	0.704	0.719
Root Mean Squared Error (RMSE)	6217.802	0.276	8.371	0.719	0.770

The error metrics from Azure and GCP are tabulated in the tables 3.2 and 3.3. For the five datasets Mean Absolute Error, Mean Squared Error, Mean Squared Log Error and Root Mean Squared Error are calculated here. Table 4 shows the average MAE and RMSE error rate between Azure and GCP, where the error values are lowest for GCP which makes this platform better performing than the other.

Table 3. 3 Error metrics of Google Cloud Platform

Datasets					
Error Metrics	Insurance	Iris	Real Estate Home Price	Wine Quality Red	Wine Quality White
Mean Absolute Error (MAE)	3820.149	0.242	6.131	0.497	0.585
Mean Squared Error	30712073	0.090	77.132	0.395	0.555
Mean Squared Log Error	0.528	0.002	0.064	0.009	0.012

Median Absolute Error	2143.254	0.209	4.97	0.391	0.496
-----------------------	----------	-------	------	-------	-------

Table 3. 4 Average Error Metrics of Azure and GCP

Platform	Average MAE	Average RMSE
Azure	1440.955	1245.587
GCP	765.521	1124.004

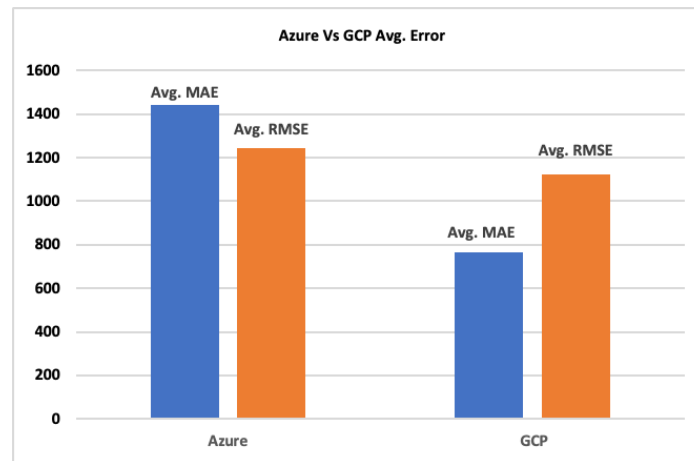


Figure 3. 15 Comparison Graph of Avg. Error between Azure and GCP

3.4. DISCUSSION

The evaluation criteria include calculating the R-Squared values and error metrics to understand how well the ML models are performing across platforms. From our experimental results in the three platforms: Azure, AWS and GCP, the average R-Squared value was found in GCP, 0.57, which is higher than the AWS and Azure. As the higher R-Squared values indicate better fitness of the ML models, it can be said that the best regression model performance is obtained in GCP. AWS is in the second position having a R^2 value 0.565. On the other side, Azure has a comparatively lower R^2 value (0.557) than the others which is open to interpretation. While comparing the error metrics among AWS and GCP, it is noticeable that the average record of error metrics in GCP (average MAE = 765.521, average RMSE = 1124.004) are comparatively lower than Azure (average MAE = 1440.955, average RMSE = 1245.5876) which denotes that linear regression model performance in GCP is better than Azure. As a continuation of the current study, we are

keen to develop further analytical and comparative study on other ML models to scrutinize more on the performance of different cloud vendors in terms of Machine Learning collaboration.

3.5. CONCLUSION

Cloud technology is making the way for enterprises to perform their technological operations more efficiently and effectively. Cloud platforms are utilized tremendously when working with ML models. However, the leadership is very competitive among the cloud vendors, i.e., Amazon Web Service (AWS), Microsoft Azure and Google Cloud Platform (GCP). It is always hard to select the -performing provider, but it is still a strong recommendation to understand the machine learning algorithms' performance before choosing any cloud vendor to work with. From the experiment conducted in the three big cloud giants, we have a clear idea of regression model performance in these platforms, and it reveals that the performance variation is not much for the three, still, it is slightly higher in GCP than the other two. Our findings will assist enterprises in understanding the performance variations of machine learning services while selecting a cloud platform to work on. Further study on other models, like, classification can be conducted and compared among the cloud platforms to achieve more insights on machine learning model performance.

Chapter 4: **Study B-** An Improved Transformer-based Model for Detecting Phishing, Spam and Ham Emails: A Large Language Model Approach

4.1. INTRODUCTION

Phishing and spam emails are pervasive and cost business resources, both time and money. This fraudulent endeavor attempts to deceive individuals into revealing sensitive and confidential information, such as financial details or login credentials. More concerning are the cyber-security implications as many breaches and attacks originate via social engineering. Threat actors use social engineering to gain an entry point via human manipulation and as a platform to launch their attacks. The majority of ransomware attacks have been linked to entry from social engineering. While Artificial Intelligence (AI) approaches have attempted to assuage these issues, heuristic-based systems continue to dominate. Radical new approaches are necessary and have emerged due to technological advances and increased research investment by both the public and private sector. The recent advancements in AI-based solutions have led to the development of innovative and unconventional strategies to combat spam and phishing tactics.

Transformer-based models have a revolutionary impact on developing spam and phishing classification models while processing, understanding, and interpreting the text data inputs. For email-based datasets, such models are continuously evolving providing additional opportunity to address the detection challenge. Furthermore, attention-based mechanisms in transformer allows model interpretability, improving the understanding of classification decisions. Large Language Models made famous by Open AI's ChatGPT, have emerged triumphant in solving new problems while being adapted to well-established challenges such as phishing and spam [64, 65]. Open AI's ChatGPT runs on its GPT engine and has made large strides in consumer and business adoption. The most famous competing LLMs are available from a plethora of vendors such as Google, Meta, and MIT while the emergence of competing LLMs such as Llama and Bert have been open sourced thereby fueling research and development from large institutions all the way down to the consumer. While these models are available for download, the ability to run pre-trained models such as BERT is still in nascent stages. As more consumers have access to local GPU technology as well as organizations like Google with Collaboratory and Hugging Face with its transformer's library and model hosting the options for implementation of applications have improved.

LLMs are general, pre-trained by the creators, and published for commercial and non-commercial licenses. There are a multitude of inputs to train a LLM such as web scraping, document corpus, and even text sources such as email and transcribed books, discussions, or speeches. While LLMs perform well on general

tasks, they can be fine-tuned to improve their performance on more specific tasks. One such example is FinBERT where BERT (Bidirectional Encoder Representations from Transformers) was trained on financial-specific documents and is able to better respond to use prompts on financial applications. Other such advances are in progress for medical data to aid in both physician decision making and end-user queries. BERT employs a self-attention mechanism that enables the model to capture contextual information and dependencies among words in any text sequence. Through the self-attention method, the weights of relevant important words are calculated. Attention scores are measured for all words or input tokens and passed through SoftMax function. A rich contextual embedding can be generated by BERT-based models which allow to excel in several natural language understanding tasks.

Within the family of BERT-based models, DistilBERT and RoBERTA are two promising variants and have been used for tasks such as fake news detection or to make predictions via Twitter data. Both models are built based on a transformer architecture and excel in NLP processing tasks. DistilBERT is designed for reducing the number of parameters making it faster and smaller version of BERT whereas Roberta is considered a more optimized and robust version. In this work, we aimed to leverage LLMs' powerful natural language processing capabilities to accurately classify and distinguish between these different types of emails. We present an Improved Phishing and Spam Detection Model (IPSDM), a custom trained and fine-tuned version of DistilBERT and RoBERTA. The issue of spam, ham (legitimate), and phishing email detection have been addressed here by developing this fine-tuned model specifically on phishing, spam, and ham data from multiple sources. We demonstrate that our fine-tuned IPSDM outperforms basic BERT and RoBERTA on both imbalanced and balanced datasets of phishing, spam, and ham.

The contribution of our work is to demonstrate a new application of LLM technology to a common problem plaguing business and society, phishing, and spam. We illustrate how an LLM can be used to approach this, and we demonstrate how fine-tuning an existing model can improve performance. This is an important step towards the application of LLMs on a large range of challenges. As LLM technology improves, our methods can be applied to improve the performance of more advanced LLMs as they are released. The rest of the paper is outlined as follows; section 4.2 explains the proposed model's framework and methodology. In section 4.3, the experimental outcomes and results are broadened. Furthermore, section 4.4 encompasses an elaborated discussion of the results. Finally, section 4.5 holds this work's concluding remarks and future prospects.

4.2. METHODOLOGY

In this paper, transformer-based self-attention mechanism models are explored to improve the pre-trained baseline BERT models. Our collected and prepared dataset is used to develop and compare models in two

different settings. 1) DistilBERT and RoBERTA were pretrained using both imbalanced and balanced phishing-ham-spam dataset, and 2) the base models' training process has been improved through applying optimization and fine-tuning mechanism. We named our proposed model as Improved Phishing Spam Detection Model (IPSDM). This model's classification performance is compared with the baseline models (DistilBERT and RoBERTA). At the end of the experiment, IPSDM exhibited substantial improvement in performance both for balanced and imbalanced scenarios compared to baseline models while detecting phishing and spam emails and texts. The top-level methodology of this research is presented in figure 4.1. Later, the breakdown of detailed flow diagram of model optimization and fine-tuning are illustrated in figure 4.5 and 4.8 of section 4.2.6.

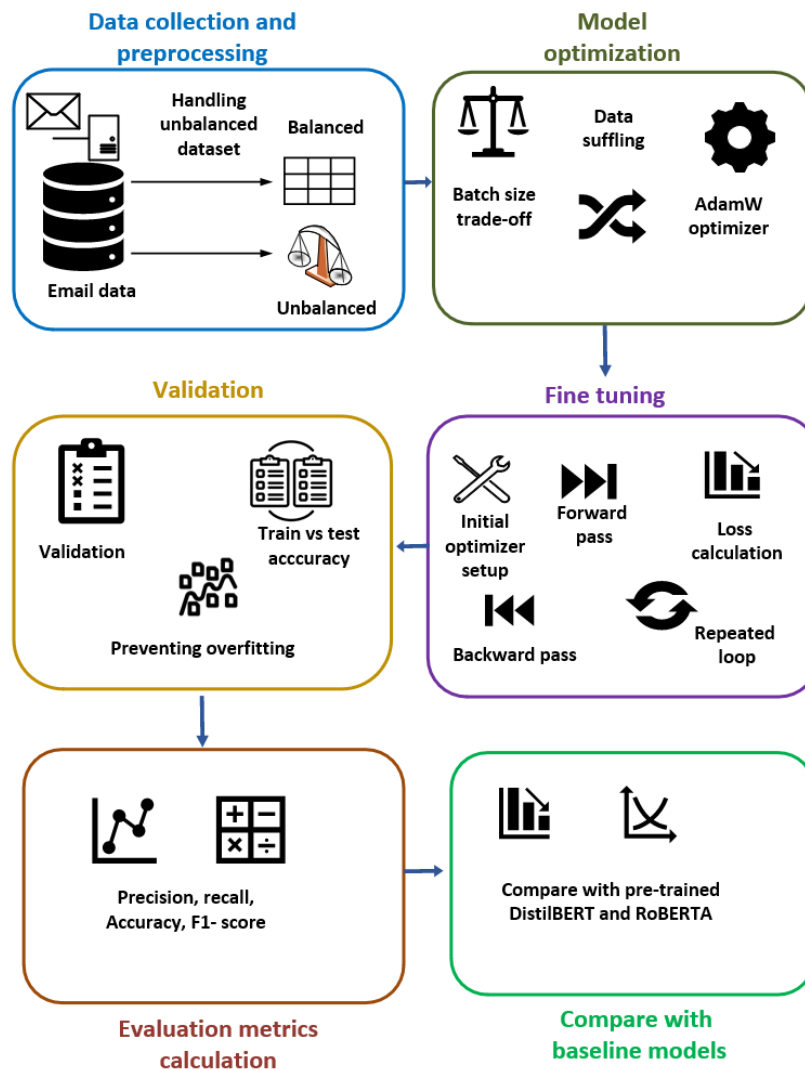


Figure 4. 1 Overall Methodology

4.2.1. Data Collection and Preparation

The data for training, testing, and validating this experiment is developed by concatenating two opensource data sources [41,42]. One dataset has ham and spam emails which is merged with another phishing email dataset. The three categories of data have been explained in the following section:

Category 1: Spam Emails- These are unsolicited messages sent in bulk to a large number of recipients, typically for advertising or fraudulent purposes. These emails often contain irrelevant or misleading content, including links to malicious websites or scams. Spam emails can clog up email inboxes, waste time and resources, and pose security risks to recipients by exposing them to potential malware or phishing attacks.

Category 2: Ham Emails- Ham emails refer to legitimate, non-spam messages that are relevant and solicited by the recipient. These emails can include personal or professional correspondence, newsletters, notifications, and other legitimate communications. While ham emails are not inherently harmful, the presence of spam and phishing emails can make it difficult for recipients to distinguish between legitimate and malicious messages, leading to potential security breaches or loss of trust in email communication systems.

Category 3: Phishing Emails- Phishing emails are fraudulent messages designed to deceive recipients into revealing sensitive information such as usernames, passwords, credit card numbers, or personal details. These emails often mimic legitimate communications from trusted sources, such as banks, social media platforms, or government agencies, in an attempt to trick recipients into clicking on malicious links, downloading malware, or providing confidential information. Phishing emails can lead to identity theft, financial fraud, data breaches, and other serious consequences for individuals and organizations.

The concatenated dataset has 747 spams, 189 phishing, and 4825 ham samples which is highly imbalanced. Such imbalanced datasets can adversely affect the performance of machine learning models, especially in terms of accuracy, precision, and recall. By balancing the class distribution through sampling, models can better learn from and accurately classify instances from all classes, leading to improved performance metrics. Moreover, the class imbalance situation can lead to biased models that perform poorly on minority classes. Sampling techniques help address this issue by either oversampling the minority class, under-sampling the majority class, or generating synthetic samples to balance the class distribution.

Here, the initial dataset has been further resampled following adaptive synthetic sampling (ADASYN) technique where minor classes (ham and spam) are oversampled by generating synthetic samples with a focus on difficult-to-learn instances. This process reduces the bias towards the majority class, making the overall predictive model more accurate and efficient. This versatile technique of sampling assists in mitigating the risk of overfitting as well. Figure 4.2 presents the feature distribution before and after

sampling. ADASYN has been preferred for its adaptability, targeted synthetic sample generation, and ability to improve model performance on imbalanced datasets while mitigating the risk of overfitting. Figure 4.3 shows a snapshot of the final dataset.

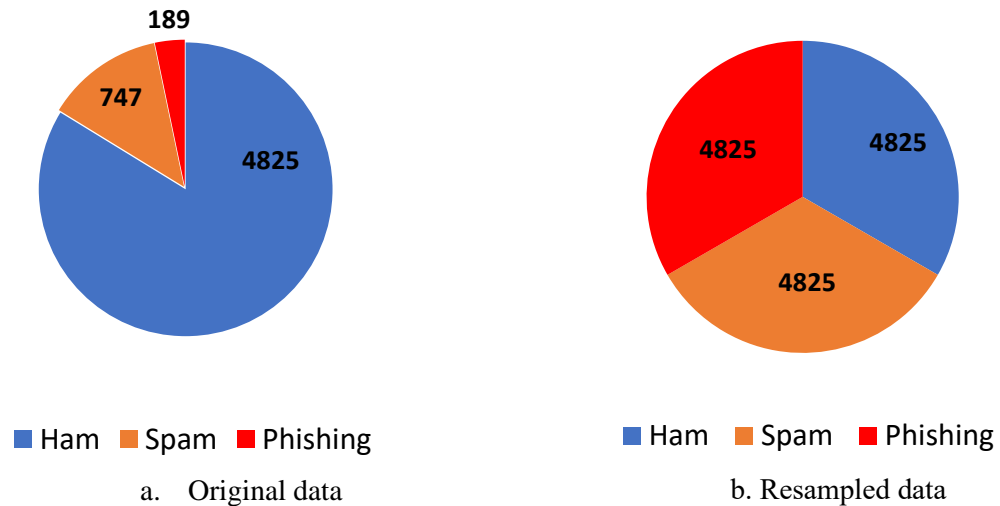


Figure 4. 2 Feature Distribution

Email	Category
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	ham
Ok lar... Joking wif u oni...	ham
U dun say so early hor... U c already then say...	ham
Shop till u Drop, IS IT YOU, either 10K, 5K, √•~£500 Cash or √•~£100 Travel voucher, Call now, 09064011000. NTT PO Box CR01327BT fixedline Cost 150ppm mobile vary	spam
Nah I dont think he goes to usf, he lives around here though	ham
refund confirmation	phishing
Even my brother is not like to speak with me. They treat me like aids patent.	ham

Figure 4. 3 A Snapshot of Dataset Overview

4.2.2. Data Splitting

The overall dataset is split into 80% (training set) and 20% (testing set). Later, from the 80% set, 60% kept for training and 20% for validation. This 20% validation set is used after the completion of each training

epoch which aids in identifying the optimal model performance. It is an integral part of the development process that ensures the model's effectiveness on unseen data identification and prediction.

4.2.3. DistilBERT

DistilBERT is a derivation of Bidirectional Encoder Representations from Transformers (BERT), which is a transformer-based model that is pre-trained for developing natural language processing tasks. The idea here is to compress the original model to make it more computationally efficient and faster. The models can be further finetuned for any specific downstream tasks on any customized dataset. DistilBERT model achieves the compression by mimicking a teacher-student model where the customized model is trained. The input tokens are the raw text inputs that need to be preprocessed. The tokenizer uses a vocabulary to tokenize the input words into sub-words. Later, the tokenized inputs are mapped to numerical embedding. The relationships between the words are captured through the attention layer. This attention mechanism works by calculating the attention score between tokens inside a sequence allowing the model to focus more on the significant relevant words than the irrelevant ones. The pooling section indicates that the entire input sequence has a fixed representation. The classifier head can be modified for any specific task, and the final prediction layer will predict the corresponding model output. For our case, this is detecting spam/ham/phishing emails.

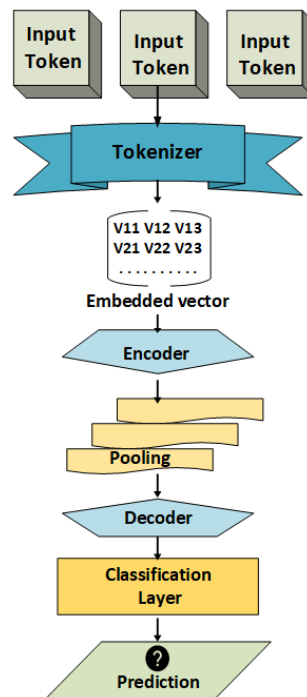


Figure 4. 4 Basic Architecture of DistilBERT and RoBERTa

4.2.4. RoBERTa

A Robustly Optimized BERT Pretraining Approach (RoBERTa) is an extended version of the transformer-based model, BERT, where model can operate on large batch size and train longer sequences. The pretraining process follows improved bidirectional context-oriented mechanism while learning the masked-out tokens for longer sequences. The architecture is similar as DistilBERT having transformer encoder layers with multi-head attention mechanisms. However, model has a byte-level tokenizer which is different than BERT. The dynamic masking works at different epochs and uses BPE as a subunit, not as characters. RoBERTa receives tokens as inputs and a tokenizer preprocess these. It passes through encoding, pooling, decoding and attention mechanism. The basic architecture of the DistilBERT and RoBERTa model is similar which is illustrated in figure 4.4.

4.2.5. Improving the Training Process

Employing the phishing-ham-spam dataset, base models of DistilBERT and RoBERTa were first measured. We aim to improve the model performance and efficiency through optimization, i.e., learning rate scheduling, adjusting batch size, sequence length and loss function, hyper parameter tuning, early stopping and fine tuning. Necessary measures have been taken to handle overfitting issue. At the end of the process, it was demonstrated that the accuracy achieved was not affected by overfitting. This proposed methodology is also employed on imbalanced dataset which was collected initially. A noteworthy improvement is observed while developing models with imbalanced dataset as well.

4.2.6. Model Optimization

The preprocessed final phishing dataset is tokenized using Hugging Face Transformers tokenizer. A sub-word-based approach is utilized by this tokenizer while breaking down the text into small unit. This allows the model to acknowledge the meaning and context of the words. The pre-trained DistilBERT and RoBERTa models are initialized with their respective pre-trained weight obtained from the pre-training process. The batch size is set 32 for training data and 64 for the validation data while trading off between memory consumption and training speed. This choice balances memory consumption and training speed, ensuring efficient utilization of computational resources while maintaining model performance. Training data is shuffled in each epoch to ensure the model's visibility of the different unseen data. This will help memorizing the training dataset and mitigating overfitting issues.

Moreover, in the optimization stage of our model training pipeline, another significant technique has been employed to enhance the learning process and prevent overfitting: choosing an optimization function. In this work, AdamW (Adam Weight Decay), an efficient optimization algorithm is used to update the weights of pre-trained models. This algorithm computes the adaptive learning rate for each parameter by combining exponential moving gradient averages and root mean square gradients. It adapts the learning rate for each

parameter in the network [66]. This means it can use different learning rates for different parts of the model, which can help speed up training and improve performance.

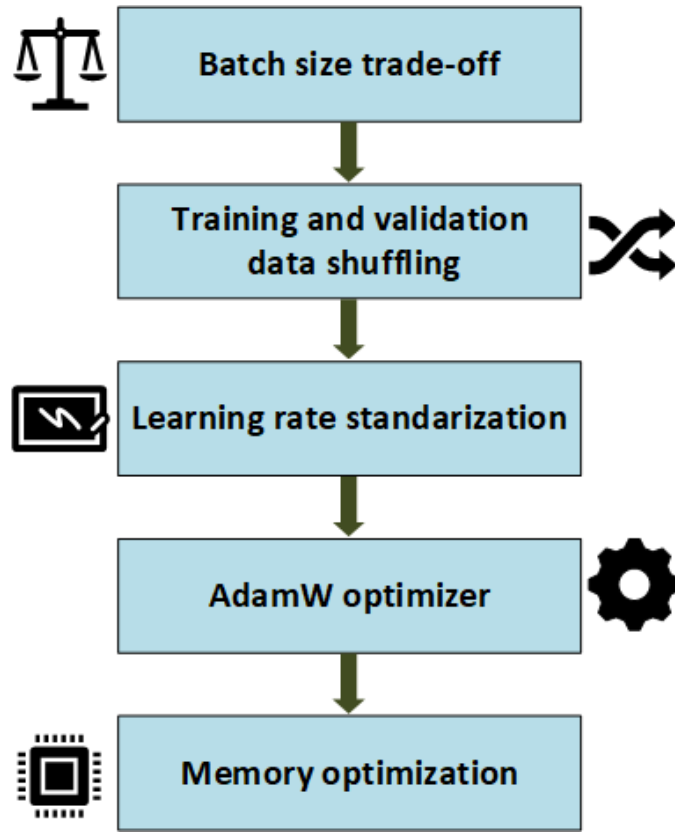


Figure 4. 5 Model optimization

L2 regularization, a weight decay mechanism, adds a penalty to the loss function, which is proportional to magnitude squared weights. This promotes the model to utilize small weights and mitigate overfitting risk by reducing the complexity of the acquired parameters. The model's parameter, Z is initialized with exponential decay rate, β_1, β_2 and ϵ with a very small value preventing division by zero. Initially, the first moment, $m_0 = 0$ and second moment, $v_0 = 0$. In each iteration, the gradient loss is calculated as below,

Gradient loss, $g = \nabla_z L(z)$.

Then, the first moment is updated, $m_i = \beta_1 * m_{i-1} + (1 - \beta_1) * g$

The updated second moment, $v_i = \beta_2 * v_{i-1} + (1 - \beta_2) * g^2$

Later, first and second-moment bias get corrected,

$$\widehat{m}_i = \frac{m_i}{1 - \beta_1^i}$$

$$\hat{v}_i = \frac{v_i}{1 - \beta 2^i}$$

Finally, the parameters are updated using the AdamW updating rule,

$$Z_i = Z_{i-1} - \frac{\text{learning rate}}{\sqrt{\hat{v}_i} + \varepsilon} \cdot (\hat{m}_i + \text{weight decay} * Z_{i-1})$$

This weight decay regularization process assists in controlling the growth of parameter values during the training, mitigating the risk of overfitting.

In the context of loss function, as this is a multiclass classification task, Cross-Entropy Loss is used which combines both SoftMax activation and negative log likelihood into a single loss term. The difference between ground truth label and probability is measured here to minimize the loss during the training process. PyTorch provides cross-entropy loss implementation that handles SoftMax computation and logarithmic computation. For a single training epoch, the loss can be defined as follow,

$$Loss_i = - \sum_{k=1}^n Z_{i,k} * \log(p_{i,k})$$

Here, $Z_{i,k}$ is the ground-truth label and $p_{i,k}$ is predicted probability made by the model.

The cross-entropy loss for the overall training is the average of individual loss,

$$Loss_{total} = 1/n \sum_{k=1}^n Loss_i$$

The models output logits for each of the class, which is passed through a SoftMax activation function for converting them into class probability. The predicted probability $p_{i,k}$ is computed as below, where $Z_{i,k}$ is the produced logit value.

$$p_{i,k} = \frac{e^{Z_{i,k}}}{\sum_{m=1}^k e^{Z_{i,m}}}$$

The optimization process diagram is presented in figure 4.5.

4.2.7. Learning Rate

An ideal learning rate for model optimization and fine-tuning depends on several factors, including model architecture, optimization algorithms and the specific task domain. It is a crucial parameter which controls

step size during the optimization process. A high learning rate might lead the model in unstable mode, resulting in poor performance for unseen data. Again, lower learning rate can slow down the convergence process. The training process might require more epochs for achieving a good result resulting in higher computational cost. An ideal learning rate graph is presented in Figure 4.6.

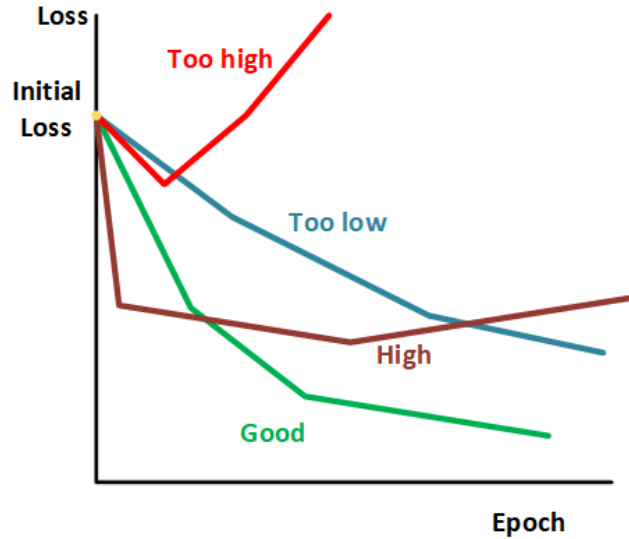


Figure 4. 6 Learning rate

In our experiment, a commonly accepted learning rate, $2e-5$ (0.00002) is set, which is standard for BERT based models, i.e., RoBERTA and DistilBERT. Later, we plot validation vs test accuracy comparison to demonstrate the effectiveness of the selected learning rate.

4.2.8. Fine Tuning

Fine tuning involves adapting a pre-trained model to get trained on specific tasks and datasets. This enhances the ability of any pre-trained NLP model to perform any domain-specific task, i.e., email classification for our case. The models are finetuned using training dataset, the 80% of the data which was separated beforehand. Training data is passed in each epoch as batches through the models, calculating the gradient using backpropagation method. To facilitate an efficient batching, DataLoader is used during the training. Code snippet is attached for RoBERTA model in figure 4.7. A similar approach is employed for DistilBERT as well. Train_loader is configured for creating mini batches of size, 32, which promotes parallel processing and optimize the memory use. Val_loader is designed to batch of 64 samples for validation ensuring most efficient evaluation method without shuffling the data. RobertaForSequenceClassification class is used to adapt the pre-trained model for specifically email classification task. This class enables an additional classification layer for the target label prediction. The overall fine-tuning process flow diagram is illustrated in figure 4.8.

```

train_dataset = EmailDataset(train_df, tokenizer)
val_dataset = EmailDataset(val_df, tokenizer)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=64, shuffle=False)

model = RobertaForSequenceClassification.from_pretrained('roberta-base', num_labels=3)

optimizer = AdamW(model.parameters(), lr=2e-5)
num_epochs = 3

```

Figure 4. 7 Code snippet of RoBERTA model DataLoader

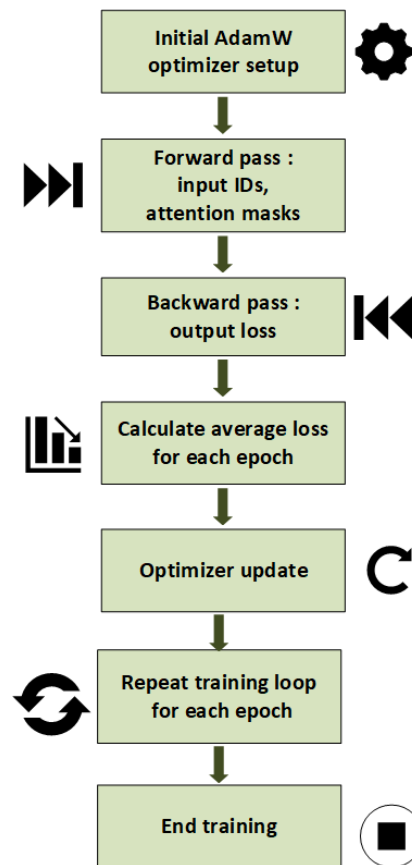


Figure 4. 8 Fine tuning process flow

4.3. RESULTS

The proposed IPSDM model is validated and tested using both unbalanced and balanced datasets. The IPSDM result metrics are compared to baseline modes, i.e., pretrained DistilBERT and RoBERTA models.

Various key metrics, including overall accuracy, precision, recall and F1-score, are calculated to assess the performance more comprehensively. These provide crucial insights of the model performance.

4.3.1. Evaluation metrics

A. Precision

The ratio of true positive predictions and the total number of positive predictions is called precision. It indicated how many predicted positive samples made by the model are actually positive. The formula for precision is as follow,

$$Precision = \frac{TP}{TP + FP}$$

High precision value suggests that the model's predicted positive instance rate is truly positive and correct. Whereas low precision indicates about making many false positive errors by the model.

B. Recall

Recall is the measurement of model's sensitivity for understanding true positive rate. It presents the ratio of true positive instances which is predicted as positive by the model. The formula for calculating recall is stated below,

$$Recall = \frac{TP}{TP + FN}$$

Higher recall conveys that the model can successfully predict the positive samples as positive making a little false negative error. However, low recall suggests that a higher number of actual positive samples are getting missed while the model predicts the false negatives.

C. F1- Score

This is a statistical metric which is the average of precision and recall which balances these values. This provides a comprehensive view of how a model deals with imbalanced datasets by trading between precision and recall. If either of precision or recall is low, then the overall F1 score will be lower. This metric validates the model's ability for predicting the positive rates and how many instances are actually positive. The formula is as follow,

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

D. Accuracy

Accuracy is the ratio of accurately predicted samples to the total number of samples made by the model[67]. It is calculated by following formula,

$$Accuracy = \frac{TP + TN}{Total\ samples}$$

However, some notable points need to consider carefully when interpreting the model accuracy because it suffers from some limitations while dealing with imbalanced data. Feature distribution across all the classes is required to be observed meticulously. Otherwise, it might raise a biased classification result. Hence, in this study, all of the essential metrics are calculated and combined together to interpret our proposed IPSDM results after running a vigilant examination.

4.3.2. Imbalanced Dataset Results

This experiment was initially carried on imbalanced datasets to assess IPSDM model's performance on imbalanced dataset. The initial collected dataset was highly imbalanced having a majority class, ham (Figure 4.2). Comparison tables (Table 4.1 and 4.2) and graphs (Figure 4.9 and 4.10) between baseline model's performance and IPSDM model's performance clearly reflect that IPSDM has a better performance in the imbalanced setting. Although the model performance is biased towards 'ham' class due to the highly uneven distribution of data samples across the three classes, it has achieved comparatively higher values than the baseline models.

Table 4. 1 Baseline DistilBERT vs IPSDM Performance (Imbalanced Dataset)

Evaluation Metrics	Base DistilBERT	IPSDM
Validation Accuracy	30.28%	51.32%
Test Accuracy	31.60%	53.67%
Validation Precision	0.841	0.972
Test Precision	0.852	0.981
Validation Recall	0.302	0.561
Test Recall	0.311	0.582
Validation F1-Score	0.432	0.613

Test F1-Score	0.451	0.621
---------------	-------	-------

Figures 4.9 and 4.10 show that the precision values are higher than recall for both cases (DistilBERT and RoBERTA). In the context of highly imbalanced characteristics of this dataset, the model can identify the majority class, ‘ham’, however, for the model struggles for classifying the minor classes, ‘spam’ and ‘phishing’.

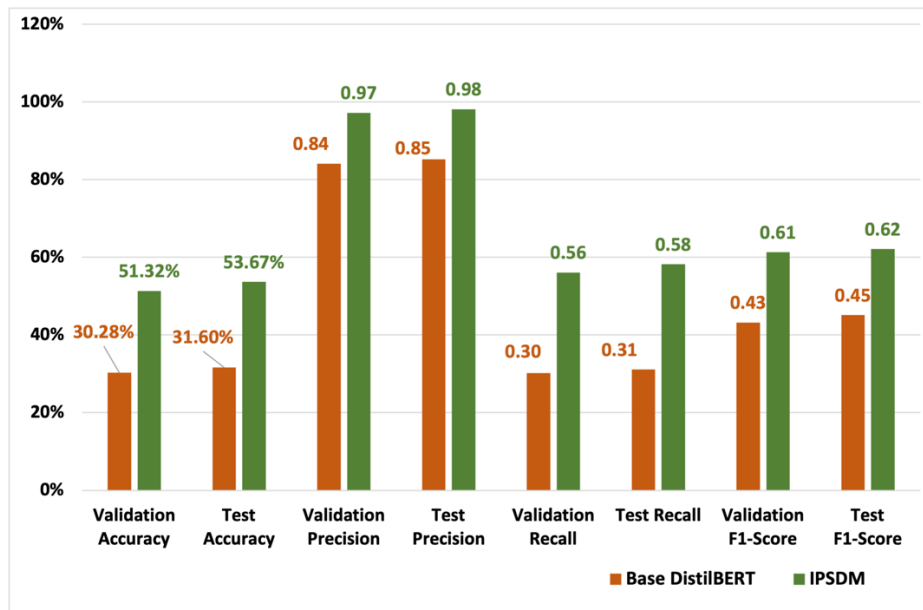


Figure 4. 9 Comparison graph of baseline DistilBERT vs IPSDM performance (imbalanced dataset)

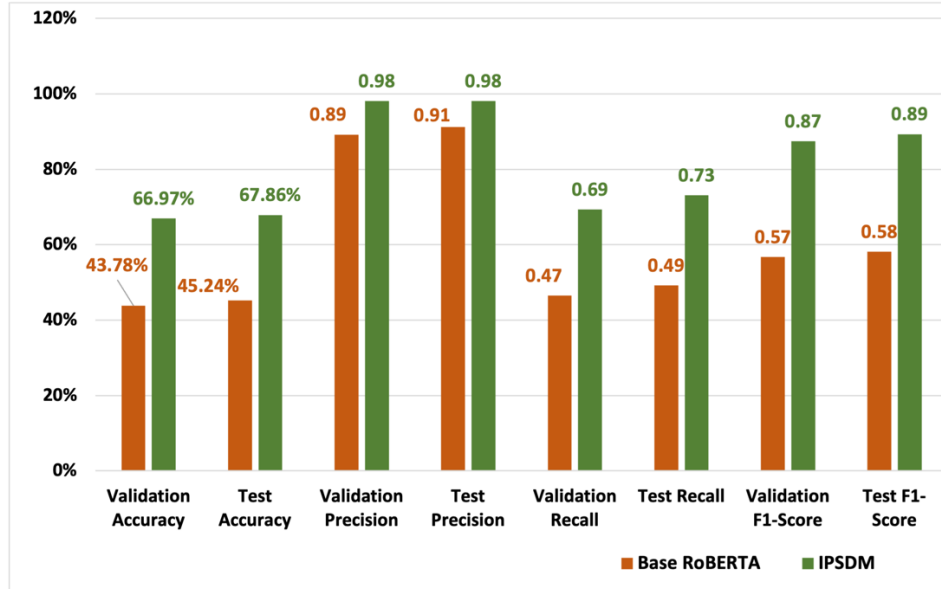


Figure 4. 10 Comparison graph of baseline RoBERTA vs IPSDM performance (imbalanced dataset)

There is a noticeable disparity between Precision and Recall values for both models. Recall values are considerably lower compared to the precision. Validation and test recall for base DistilBERT model are 0.30 and 0.31 (shown in Table 4.1). For the base RoBERTA model, the recall values are 0.47 and 0.49 (shown in 4.2). This suggests that the models are facing challenges for identifying the minor classes, ‘spam’ and ‘phishing’ due to the imbalanced nature. However, it is noteworthy that the performance of IPSDM for DistilBERT and RoBERTA is notably higher even though the dataset is imbalanced.

Table 4. 2 Baseline RoBERTA vs IPSDM performance (imbalanced dataset)

Evaluation Metrics	Base RoBERTA	IPSDM
Validation Accuracy	43.78%	66.97%
Test Accuracy	45.24%	67.86%
Validation Precision	0.892	0.981
Test Precision	0.912	0.981
Validation Recall	0.465	0.693
Test Recall	0.492	0.731

Validation F1-Score	0.567	0.874
Test F1-Score	0.581	0.893

4.3.3. Balanced Dataset Results

The collected email datasets have been resampled and balanced. After preparing this balanced dataset, baseline DistilBERT and RoBERTA models were trained and validated. Again, using a similar dataset, we worked on model optimization and fine tuning. The evaluation metrics of our proposed model, IPSDM and the baseline models are tabulated in Tables 4.3 and 4.4. Accuracy, precision, and recall for validation and test cases are presented here. Also, the values are illustrated in comparison graphs (Figure 4.11 and Figure 4.12).

Table 4. 3 Baseline DistilBERT vs IPSDM performance (balanced dataset)

Evaluation Metrics	Base DistilBERT	IPSDM
Validation Accuracy	82.63%	97.50%
Test Accuracy	88.95%	97.10%
Validation Precision	0.8543	0.9755
Test Precision	0.9025	0.9716
Validation Recall	0.6971	0.9750
Test Recall	0.7532	0.9710
Validation F1-Score	0.8867	0.9749
Test F1-Score	0.8943	0.9710

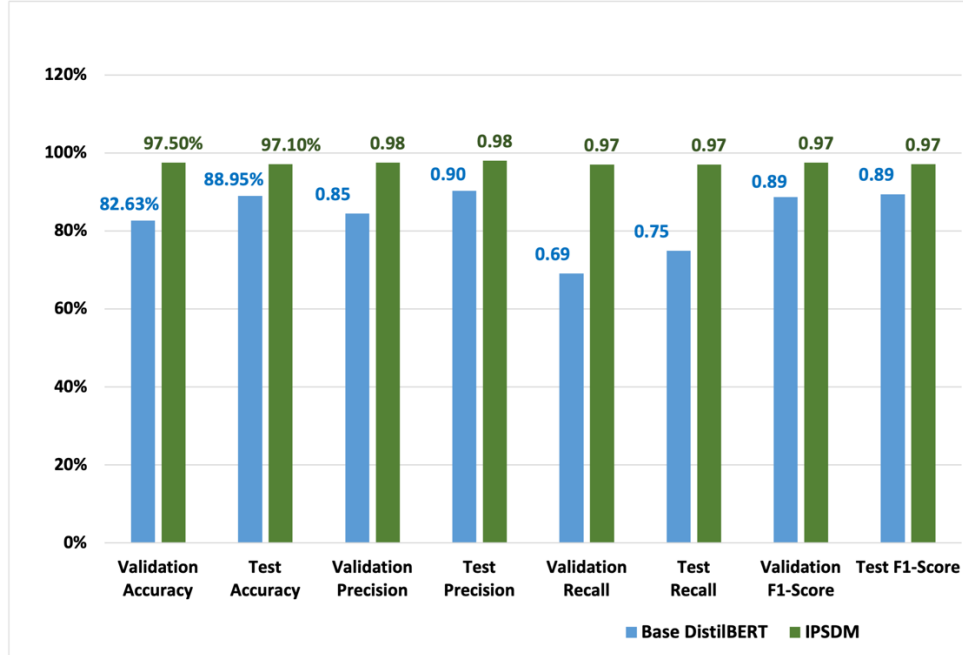


Figure 4. 11 Comparison graph of Baseline DistilBERT vs IPSDM performance (balanced dataset)

The evaluation metrics from Figures 4.11 and 4.12 exhibit an increase in validation accuracy- approximately 14.87% and 11.89%; test accuracy approximately 8.15% and 5.71% respectively for base DistilBERT and RoBERTA models vs IPSDM. A consistent rise in F1scores suggests that the IPSDM has elevated performance across both cases. This score is the harmonic mean of recall and precision, a crucial metric for assessing the balance between the crucial aspects of classification performance.

Table 4. 4 Baseline ROBERTA vs IPSDM performance (balanced dataset)

Evaluation Metrics	Base ROBERTA	IPSDM
Validation Accuracy	87.10%	98.99%
Test Accuracy	93.29%	99.00%
Validation Precision	0.921	0.982
Test Precision	0.853	0.991
Validation Recall	0.903	0.989
Test Recall	0.923	0.991
Validation F1-Score	0.911	0.982

Test F1-Score	0.931	0.985
---------------	-------	-------

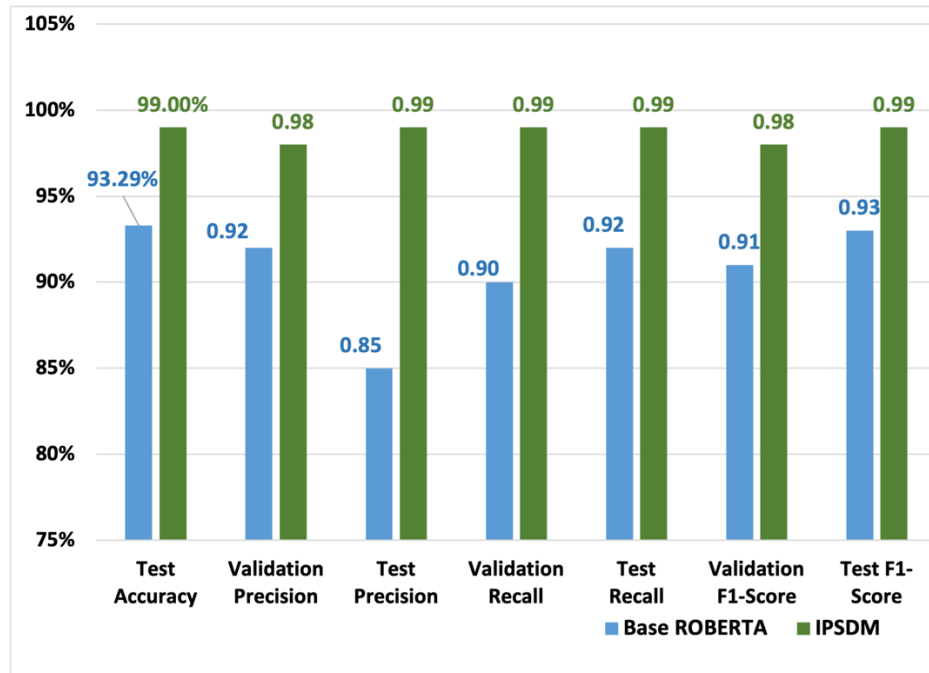


Figure 4. 12 Comparison graph of Baseline RoBERTa vs IPSDM performance (balanced dataset)

4.3.4. Avoiding Overfitting

A common issue in statistical modellings and machine learning is overfitting which occurs when a model is performs too well on the training dataset, however, too poorly on the new or unseen data, i.e., testing dataset. Overfitting can be effectively managed in balanced situations while a model has consistent performance on validation and test datasets. A close alignment between test and validation accuracy suggests that the classification models yield good results on unseen, new data. In the balanced scenario, test and validation accuracy values indicate minimal disparity, i.e., 97.10% vs 97.50% and 99. 00% vs 98.99%.

Table 4. 5 Validation vs test accuracy

Model Name	Validation accuracy	Test accuracy
Balanced_ IPSDM/ DistilBERT	97.50%	97.10%

Balanced_ IPSDM/ RoBERTA	98.99%	99.00%
Imbalanced_ IPSDM/ DistilBERT	51.32%	53.67%
Imbalanced_ IPSDM/ RoBERTA	66.97%	67.86%

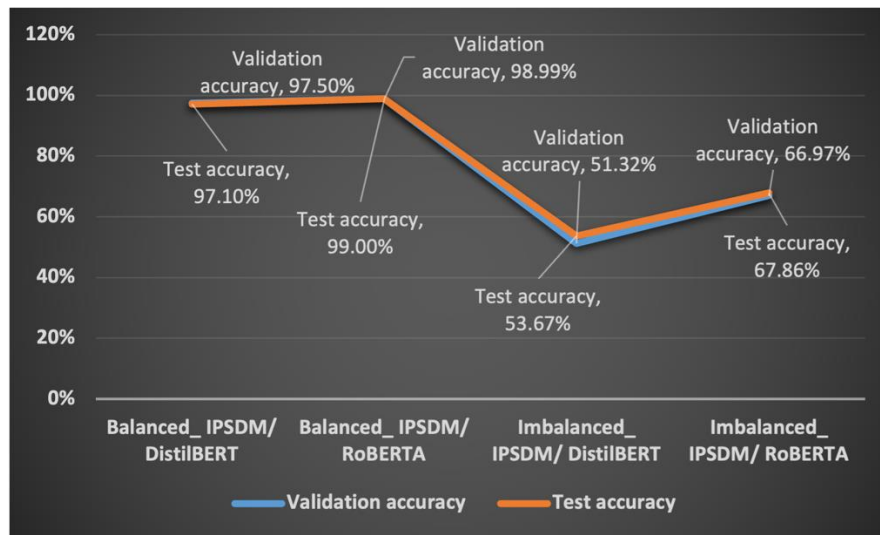


Table 4. 6 Validation vs test accuracy graph

Based on the data from Table 4.5 and Figure 4.13, there is no large gap between validation and test accuracy. When training or validation accuracy is notably higher than test accuracy, there is a high chance of overfitting. Moreover, the precision, recall and F1 measures from table 4.1 through 4.4 also suggest a harmonic distribution of these metrics which is also a positive indication. The comparison graph for both validation and test accuracy lines almost overlap with each other, indicating that the model is performing well on the unseen data.

4.4. DISCUSSION

The results from imbalanced and balanced settings depict an enhancement in performance for the IPSDM model. Validation and test accuracy are separately measured to understand if there is any overfitting issue persist. Baseline DistilBERT has 82.63% validation accuracy and 88.95% testing accuracy whereas IPSDM has 97.50% and 97.10% validation and test accuracy, respectively. The baseline model's accuracy variation

is 6 (+32) % in training and testing performance reveal that base DistilBERT is exhibiting a minor overfitting problem. However, this has been effectively handled during the development of IPSDM DistilBERT version. Again, a similar trait is visible for base RoBERTA and IPSDM for RoBERTA as well. Base RoBERTA model's validation and testing accuracy gap is around 6 (+- 19) % whereas IPSDM has 0.01% of difference between these two values.

Such evaluation has also been extended to imbalanced dataset to analyze how IPSDM performs in challenging scenarios. In the imbalanced setting, Tables 4.3 and 4.4 presents that our proposed model has outperformed the baseline models in this scenario. While the IPSDM model has demonstrated significant improvements in performance compared to baseline models, there are still areas for improvement. One limitation is the potential bias towards the majority class ('ham') due to the heavy skew-ness of the sample distribution. The precision values for both baseline and IPSDM models are notably higher which are 0.85 and 0.98 for base DistilBERT and IPSDM test precision; 0.91 and 0.98 for base RoBERTA and IPSDM respectively, present that the model is predicting most of the instances as 'ham'. This bias can result in higher precision but lower recall, indicating that the model may be overly conservative in classifying instances as 'ham'. Future research could explore techniques to balance the dataset further or modify the model architecture to better handle class imbalances. However, later applying ADASYN, an advanced sampling technique, this class imbalanced situation is handled at the initial stage. A prominent change in performance is hence demonstrated in IPSDM models both for DistilBERT and RoBERTA both for balanced and imbalanced datasets scenarios.

Our work shows how emergent large language model (LLM) technology can be leveraged to solve existing issues presented in the phishing and spam problem. While NLP and other traditional machine learning approaches are viable, using LLMs has vast potential as LLMs advance and continue to transform society. We illustrate how a LLM can be custom trained to improve results. In our case, we show that we can have an impact in improving phishing and spam detection. While the future of LLM is vast, we are at the nascent stages of applying LLMs to existing problems. Our results can help secure cyberspace and save businesses time and money by detecting phishing and spam thereby improving their cyber-defense and improving their overall cyber-health. Furthermore, our method can be extended to new emerging LLM models which are improving at an astounding pace. Future researchers can base studies by applying our method to newly emerging LLMs and apply our method to a wide array of other challenges facing business and researchers.

4.5. CONCLUSION

Solving long-standing societal issues via radical new approaches, specifically LLMs, shows great promise to improving the lives and experiences of computing users the world over. Phishing and Spam have long

since been an issue causing lost time and straining the financial resources of consumers and organizations. We demonstrate how leveraging new technology can be applied to these persistent challenges. LLMs offer society great benefits and we have only scratched the surface on their potential. In the future, improving the quality of life via multiple dimensions will be realized such as medical diagnoses, chat-bots, education, and security to name a few. This work demonstrates how LLMs can be leveraged to detect phishing and spam by leveraging LLMs and then presenting our fine-tuned version, IPSDM. Following the proposed mechanism, modified DistilBERT could achieve 97.50% of validation and 97.10% of test accuracy with a F1-score of 0.97. Again, the modified RoBERTA model obtained 98.99% of validation and 99.00% of test accuracy, including a F1-score of 0.98. The result of this study presents the effectiveness of IPSDM model while reducing the overfitting issues and handling imbalanced datasets. The attained accuracy has surpassed the existing state-of-the-art models.

While the IPSDM model has shown promising results in both balanced and imbalanced dataset scenarios, further evaluation and validation are necessary to ensure its robustness and generalizability across different datasets and settings. This could involve testing the model on larger and more diverse datasets and conducting cross-validation experiments to assess its performance under various conditions. Future work entails further refinement of IPSDM via incorporation of additional tuning techniques as well as hyperparameter tuning and combining with ensemble modeling. Applying data augmentation such as text rotation, contrastive learning, and synonym replacement might also increase the diversity and improve the training performance. Furthermore, the field of Large Language Models has attracted substantial investment from industry and consumers causing it to develop rapidly with new open-source models being released nearly daily. We aim to experiment with further LLMs such as Meta's Llama and Llama 2. Infusing such solutions into chatbot, web applications and other real-world practical systems would serve society in numerous valuable ways.

Chapter 5: **Study C-** Perception and Evaluation of Text-to-Image Generative AI Models: A Comparative Study of DALL-E, Google Imagen, GROK, and Stable Diffusion

5.1. INTRODUCTION

The emergence of generative AI frameworks has transformed the landscape of the creation of digital image creation, marking a new era of AI to generate diverse and realistic images from text prompts. Text to image generative models are AI models that leverage Natural Language Processing (NLP) techniques while interpreting the textual inputs to visual representation [68]. Such generative models enable users to articulate ideas and visual concepts thorough natural language, offering numerous opportunities for various applications, i.e., entertainment, art, design, and visual communication. Some renowned text-to-image generation models include DALLE, Google Imagen, GROK, and Stable Diffusion. Although deepfake technology has garnered considerable attention and scrutiny, recent advancements in text-to-image generative models have unlocked new opportunities in creative fashion and practical applications. However, alongside the remarkable potentials of AI-generated images, challenges and considerations exist regarding their quality, realism, and societal impact.

Text-to-image synthesis, the process of generating images from textual prompts, offers both promises and complexities. On one hand, this technology holds potential across various domains such as design, art, entertainment, and visual storytelling. On the other hand, ensuring the accuracy, coherence, and ethical implications of AI-generated images presents significant challenges. The reliance on textual prompts introduces complexities in accurately conveying desired visual concepts, leading to potential discrepancies between the intended and generated images. Furthermore, other concerns such as misinformation, bias, and manipulation highlight the significance of rigorous evaluation techniques and safeguards in developing and deploying text-to-image synthesis systems.

While mathematical methods provide quantitative assessments of text-to-image synthesis, metrics such as the Fréchet Inception Distance (FID), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) offer objective benchmarks for evaluating image fidelity and similarity to real-world counterparts. These metrics employ computational algorithms to assess various aspects of image quality, including perceptual similarity, structural integrity, and noise levels. By quantifying these attributes, mathematical metrics offer valuable insights into the technical performance of AI-generated images and enable comparisons across different models and datasets.

However, despite their utility, mathematical metrics have inherent limitations. Mathematical algorithms may not fully capture the subjective nature of human perception. Human observers possess the ability to

discern subtle details, interpret contextual cues, and make qualitative judgments that extend beyond numerical measurements. Thus, human evaluation remains indispensable in the assessment of image realism and perceptual fidelity based on certain criteria such as contextual coherence, emotional resonance, and aesthetic appeal. By incorporating human evaluation alongside mathematical metrics, researchers can validate the technical accuracy of AI-generated images while contextualizing their perceptual impact within real-world contexts. In essence, the integration of mathematical metrics and human evaluation represents a synergistic approach to assessing the quality and realism of text-to-image synthesis.

This study evaluated four popular AI tools, including DALL-E, Google Imagen, Stable Diffusion, and GROK AI, focusing on their text-to-image diffusion models. Ten real images were collected from diverse sources to serve as benchmarks for evaluation. Our research addresses the challenges and opportunities inherent in text-to-image synthesis through a robust evaluation approach. We explored three mathematical formulas- FID, SSIM, and PSNR metrics across the image datasets, collected from four AI platforms to measure image quality and realism. Additionally, we conducted human evaluations in which participants assessed the realism and quality of AI-generated images, enabling a comparative analysis between mathematical metrics and human perception. Through this interdisciplinary approach, we aim to contribute to understanding and advancing text-to-image synthesis while promoting responsible and ethical AI development.

5.2. METHODS

This study explores text-to-image diffusion models, focusing on four prominent text-to-image generation models: DALL-E, Google Imagen, Stable Diffusion, and GROK AI. Ten real images were collected from diverse sources to serve as benchmarks for evaluation. Utilizing identical text prompts, ten sets of images were generated using each of the four platforms. The null hypothesis suggests that there is no significant distinction between the generated images and real images. To investigate this, another alternative research hypotheses has been initially formulated. That suggests that DALL-E and Google Imagen produce more realistic images than the other platforms. Two techniques were employed to evaluate the generated images: computer evaluation and human evaluation via a survey. For computer evaluation, metrics such as FID score, PSNR, and SSIM were computed for each set of generated images in comparison to the real ones. The results were analyzed and further validated through a survey involving thirty participants. The overall methodology is presented in figure 5.1.

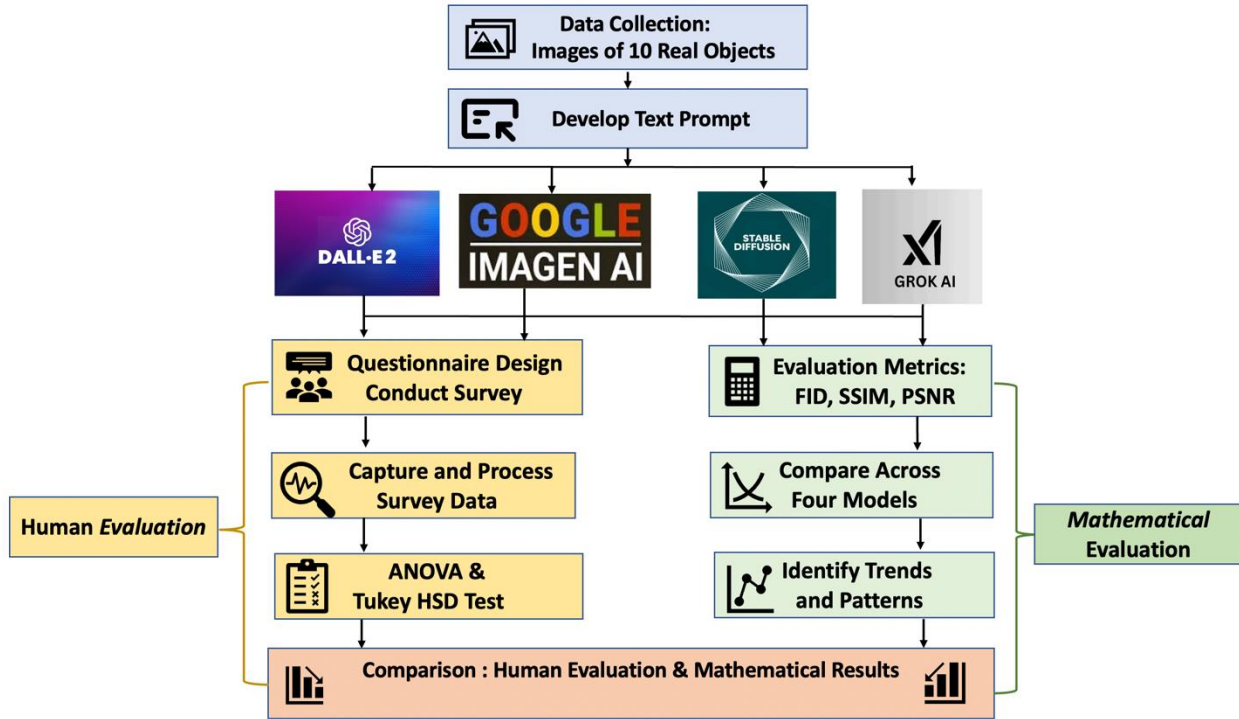


Figure 5. 1 Overall Method Flow Diagram

5.2.1. Text to image diffusion models

The diffusion process in image generation refers to a method where noise is gradually added to an initial image over multiple steps. The noise is smoothed or diffused at each step, resulting in a new image that becomes increasingly distorted or randomized [69]. This process is typically guided by a diffusion model, which defines how the noise is applied and how it evolves over time. An ideal text-to-image diffusion model follows the below general steps:

Step 1: Text Embeddings:

- At first the text descriptions are converted into a dense vector representation, which is known as text embedding, using techniques such as pre-trained language models (i.e., BERT, RoBERTA) or word embeddings.

Step 2: Diffusion Process:

- The diffusion process involves iteratively applying noise to the input text embeddings to generate intermediate noisy embeddings [70].
- These noisy embeddings are then passed through a diffusion model, which gradually improves the image features to generate more realistic images.

Step 3: Image Generation:

- The final refined embeddings are passed through a decoder network to generate images which correspond to the original textual description.
- Theis decoder network may consist of convolutional neural networks (CNNs), or other architectures tailored for particular image generation [71].

Step 4: Training:

- During this training process, the model learns to minimize the difference between the generated and ground truth images corresponding to the input text descriptions.
- The diffusion model and decoder network parameters are optimized using techniques like maximum likelihood estimation or adversarial training.

Step 5: Evaluation:

- The quality of the generated images is evaluated using metrics such as FID score, SSIM, or human perceptual studies to assess realism and coherence with the input text descriptions.

5.2.2. DALL-E

DALL-E, stands for "Distribute Aggregate Linear Latent Encoder," is a groundbreaking text-to-image generation model developed by OpenAI. DALL-E is built upon the transformer-based architecture, which is similar to the one used in the GPT series of models, to encode the textual description into a dense vector representation [72]. This text embedding serves as conditioning information for any image generation process. Unlike traditional image generation models that rely on continuous latent spaces, DALL-E introduces a discrete latent space via a Vector Quantized Variational Autoencoder (VQ-VAE-2) model. Such discrete latent space allows for the representation of diverse image features in a compact and interpretable manner.

One of the key strengths of DALL-E is producing highly diverse and semantically meaningful images based on a wide range of textual prompts. By leveraging the rich semantic representations encoded in the text embeddings, DALL-E can capture intricate details and nuances in the generated images, ranging from simple objects to complex scenes and abstract concepts. Additionally, the discrete latent space introduced by the VQ-VAE-2 model allows for fine-grained control over the generated images, enabling users to manipulate various visual attributes such as style, color, and compositions.

5.2.3. Google Imagen

Google Imagen is a text-to-image generation platform developed by Google, which leverages advanced machine-learning techniques for image synthesis from textual descriptions [73]. While specific details of the architecture are not publicly disclosed, Google Imagen likely employs transformer-based models similar to those used in DALL-E or GPT for text encoding and generation. The platform may incorporate additional components for image synthesis, such as attention mechanisms or generative adversarial networks (GANs), to enhance the realism and diversity of generated images.

5.2.4. Stable Diffusion

Stable Diffusion, introduced in 2022, stands as a prominent deep learning model within the ongoing surge of AI advancements. Utilizing diffusion techniques, it primarily functions as a text-to-image model, capable of generating intricate visuals based on textual inputs. Beyond image generation, Stable Diffusion demonstrates versatility across tasks such as inpainting, outpainting, and facilitating image-to-image translations guided by text prompts. Additionally, it offers an "img2img" sampling script, which takes a text prompt, an existing image path, and a strength value ranging from 0.0 to 1.0. This script outputs a new image derived from the original one, integrating elements specified in the text prompt. The strength parameter controls the level of noise injected into the output image, influencing the degree of variation. Higher strength values yield more diverse images but may stray from semantic coherence with the provided prompt.

5.2.5. GROK AI

GROK AI is another prominent text-to-image generation platform that utilizes deep learning techniques based on textual descriptions for image synthesis. GROK AI employs transformer-based models or similar architectures for text encoding and image generation. The platform incorporates custom components or optimizations tailored to the text-to-image generation task to enhance the quality and fidelity of generated images. GROK AI aims to provide users with a seamless experience for creating realistic images from textual prompts, leveraging state-of-the-art machine learning algorithms for efficient and effective image synthesis.

5.3. EXPERIMENT AND RESULT ANALYSIS

We collected real images from 10 subjects. Then, using a specific similar text prompt, similar images were generated using AI-based image generation tools: DALL-E, Imagen, Stable Diffusion, and GROK. Consequently, for each subject, a set of five images was compiled, comprising the original real image alongside four AI-generated images. For instance, the figure 5.2 below displays a real image featuring a cat

swimming, accompanied by four additional images generated by AI tools. This process was repeated across all subjects, resulting in a comprehensive collection of images for evaluation and analysis.



Figure 5. 2 Input Image of a Cat Sswimming

5.4. EVALUATION

Two distinct evaluation methods were employed to assess the realism of the generated images. Method A involved mathematical evaluation, where metrics such as Fréchet Inception Distance (FID) score, Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) ratios were calculated using Python frameworks and libraries. Method B consisted of a human study conducted via a survey, with data gathered from 28 subjects. Statistical evaluation was performed on the collected data, including ANOVA tests and Tukey post hoc analysis, to determine the significance of each group. These two evaluation methods will be elaborated upon in the following sections.

5.4.1. Method A: Mathematical Evaluation

The Fréchet Inception Distance (FID) Score

The Fréchet Inception Distance (FID) score is a metric commonly used to evaluate the quality of generated images in generative adversarial networks (GANs) and other image generation models. It measures the

similarity between the distributions of real and generated images in a feature space learned by an Inception-v3 neural network [74]. The FID score is computed based on the mean and covariance of feature representations of real and generated images. Given two sets of feature representations μ_r and μ_g (mean); Σr and Σg (covariance) for real and generated images respectively, the FID score is calculated using the following formula:

$$FID = \|\mu_r - \mu_g\|_2^2 + T_i (\Sigma r + \Sigma g - 2 \left((\Sigma r \Sigma g)^{\frac{1}{2}} \right))$$

Where:

- $\|\mu_r - \mu_g\|_2^2$ presents the squared Euclidean distance between the mean feature vectors.
- T_i represents the trace of matrix.
- Σr and Σg are the covariance matrices of real and generated feature representations, respectively.

The Structural Similarity Index (SSIM)

It is a metric used to quantify the similarity between two images, considering their luminance, contrast, and structure. It measures the perceptual difference between the original and processed images [75].

SSIM is calculated using three components: luminance comparison, contrast comparison, and structure comparison. The overall SSIM score is the product of these three components.

The formula for SSIM is as follows:

$$SSIM(x, y) = \frac{l(x, y) c(x, y)}{M1 M2}$$

$$\text{Here, } l(x, y) = \frac{2 \mu_x \mu_y + C1}{\mu_x^2 + \mu_y^2 + C1}$$

$$C(x, y) = \frac{2 \sigma_x \sigma_y + C2}{\sigma_x^2 + \sigma_y^2 + C2}$$

$C1$ and $C2$ are constants here and the values are chosen to avoid zero instability.

x and y are the two images that are being compared.

A code snippet of this calculation using Python programming is attached in figure 5.3.

Peak Signal-to-Noise Ratio (PSNR)

This metric is commonly used to evaluate the quality of reconstructed or compressed images. It measures the difference between the original image and its approximation, considering the difference's magnitude and the image's maximum possible range. The PSNR is calculated based on the Mean Squared Error (MSE) between the original image I and the reconstructed I' in decibels (dB). When the maximum possible pixel value is MAX , i and j are image dimensions, the formula for PSNR is as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

$$\text{Here, } MSE = \frac{1}{ij} \sum_{k=0}^{i-1} \sum_{l=0}^{j-1} [I(k, n) - I'(k, n)]^2$$

```
def calculate_frechet_distance(mu1, sigma1, mu2, sigma2):
    epsilon = 1e-6

    # Numerical stability.. tried this to handle negative value issue!
    sqrtm_term = np.real(linalg.sqrtm(np.dot(sigma1, sigma2)))

    # Handle small eigenvalues
    if np.iscomplexobj(sqrtm_term):
        sqrtm_term = sqrtm_term.real

    fid = np.linalg.norm(mu1 - mu2) + np.trace(sigma1 + sigma2 - 2 * sqrtm_term + epsilon)

    return fid

[ ] def calculate_fid(real_images, generated_images, model):
    real_mu, real_sigma = calculate_activation_statistics(real_images, model)
    generated_mu, generated_sigma = calculate_activation_statistics(generated_images, model)
    fid = calculate_frechet_distance(real_mu, real_sigma, generated_mu, generated_sigma)
    return fid
```

Figure 5. 3 Code snippet of FID Score Calculation

Necessary libraries are imported, including numpy for numerical computations, skimage.metrics for SSIM calculation, and tensorflow for PSNR calculation. A function called `calculate_metrics` is defined for calculating the metrics, which takes the real image and generated image as input. Inside the function, the FID score is first calculated using a function called `calculate_fid`. This function, assumed to be defined elsewhere, takes real and generated images along with an inception model as input. The SSIM score is then obtained using the SSIM function from `skimage.metrics`. This function computes the structural similarity between two images. Finally, the PSNR score is evaluated using the `tf.image.psnr` function from

TensorFlow. PSNR measures the quality of an image in terms of signal-to-noise ratio. These scores are provided as quantitative assessments of the realism and quality of the generated images, complementing the human evaluation conducted in the study.

The average FID, SSIM and PSNR scores are noted in table 5.1 below.

Table 5. 1 Mathematical Evaluation Metrics

AI Tool\ Metrics	FID	SSIM	PSNR
DALLE	9.00%	1.35%	9.88
IMAGEN	10.43%	0.86%	10.20
GROK	10.69%	1.50%	10.51
Stable Diffusion	15.95%	0.95%	9.21

These three values have been illustrated in the graph to get the comparison results among three AI tools. As the lower percentages of FID indicate better similarity between real and generated images, implying higher quality and realism, DALL-E has the lowest FID score (9.0%) indicating better similarity with real images. On the other hand, the FID score is higher in Stable Diffusion (15.95%) than the other three. Moreover, the higher SSIM and PSNR score mean the better similarity with the real image. In this case, DALL-E has the highest value compared to other three suggesting its generated images have better quality and realism.

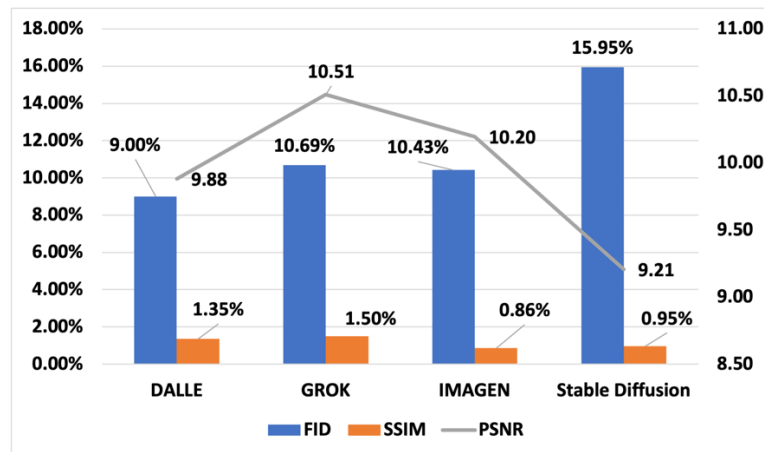


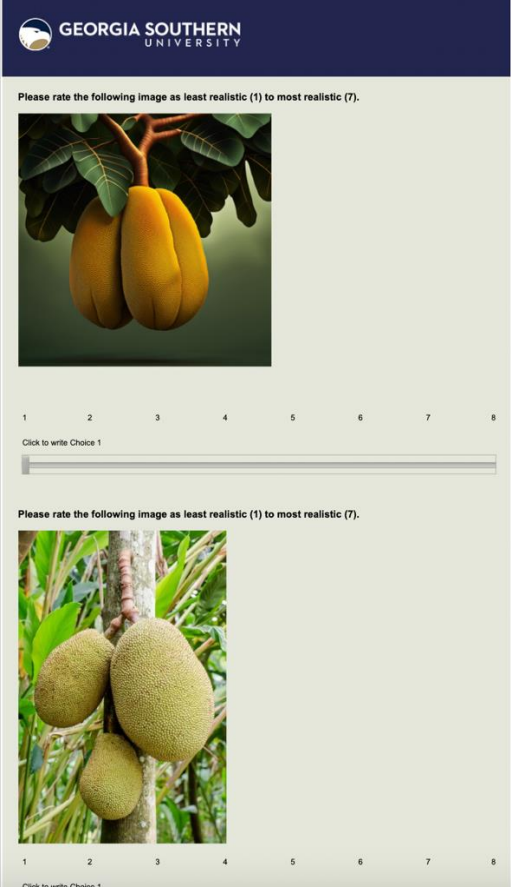
Figure 5. 4 Comparison of Mathematical Metrics

5.4.2. Method B: Human Evaluation

In this study, as a part of human evaluation, a survey was developed and administered using the Qualtrics online survey platform to gather subjective evaluations of the images generated by human participants. The participants were asked to provide ratings for image realism on a scale of 1 to 7. The survey consisted of 10 sets of questions, with each set containing five sets of images. Participants were asked to rate the realism of the images, with 1 indicating less realistic and 7 indicating more realistic. The responses collected from the survey were structured and grouped based on five categories, including real images and images generated by DALL-E, Imagen, Stable Diffusion, and GROK AI. Each category represented a different image generation model.


5.4.1. Survey Result and Statistical Analysis

A total of 28 subjects responded during the survey comprising adults aged 18 to 44, with a gender distribution of 17 males and 11 females, all of whom were citizens of the United States of America. Each participant was randomly allocated four out of ten sets of questions, providing a diverse perspective on image realism assessment across different age groups and genders. Statistical analyses were conducted using ANOVA tests and Tukey's Honest Significant Difference (HSD) test to determine the significance of differences among the groups. Figure 5.5 presents a snapshot of survey question.



GEORGIA SOUTHERN UNIVERSITY


Please rate the following image as least realistic (1) to most realistic (7).



1 2 3 4 5 6 7 8

Click to write Choice 1

Please rate the following image as least realistic (1) to most realistic (7).



1 2 3 4 5 6 7 8

Click to write Choice 1

Figure 5. 5 Survey Questions Snapshot

Analysis of Variance (ANOVA)

ANOVA is a statistical technique used to compare the means of two or more groups to check if they are significantly different from each other. ANOVA assesses the impact of one or more factors by comparing the means of different samples [76].

The formula for one-way ANOVA is:

$$F = \frac{\text{Between group variability}}{\text{Within group variability}}$$

Tukey's Honest Significant Difference (HSD) Test

This post hoc test is conducted following ANOVA to assess the significance of differences between pairs of group means [77]. It identifies which specific groups differ significantly from each other. The formula for Tukey's HSD calculation is as below,

$$HSD = \sqrt{\frac{EMean Square Error}{n}} * Critical value$$

The results of these statistical tests are tabulated in the following tables, providing insights into the significance of differences among the image generation models. Table 5.2 summarizes the one-way ANOVA test result, including each group's average and variance.

Table 5. 2 Group Summary

Groups	Count	Sum	Average	Variance
DALL-E	112	639	5.71	4.03
Stable Diffusion	112	535	4.78	4.90
Imagen	112	642	5.73	3.68
GROK	112	458	4.09	4.95
Real Image	112	650	5.80	4.11

The difference between groups from ANOVA test is tabulated in table 5.3. The Between Groups Sum of Squares represent the sum of squares of the differences between the group means and the overall mean. It quantifies the variability in ratings among the different groups of images. The value between groups is 257.619, indicating significant variability in ratings among the different image generation models.

Degrees of Freedom (df) suggests the degrees of freedom associated with each source of variation [78]. For the Between Groups source, $df = k - 1$, where k is the number of groups. For the Within Groups source, $df = N - k$, where N is the total number of observations and k is the number of groups.

Table 5. 3 Group Differences

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	257.619	4	64.405	14.856	0.000
Within Groups	2401.755	554	4.335		
Total	2659.374	558			

The F-value is the ratio of the between-groups mean square to the within-groups mean square. It indicates whether the differences between group means are statistically significant. A larger F-value suggests a greater difference between group means. This is the p-value associated with the F-statistic. It indicates the probability of obtaining the observed F-value if the null hypothesis (i.e., no differences between group means) is true. A significance value less than the chosen alpha level (e.g., 0.05) indicates that the differences between group means are statistically significant. In this case, $F = 14.856$, with a significance value of 0.000, suggests that the differences between group means are statistically significant. Therefore, we can reject the null hypothesis and conclude that there are significant differences in image realism ratings among the different image generation models.

For further understanding the variance among groups, Tukey's Honest Significant Difference (HSD) Test have been conducted. The results are noted in table 5.4. Each pair of groups is compared in this test, and the mean difference, standard error, significance level, and confidence interval are provided.

Table 5. 4 Tukey HSD Result

Multiple Comparisons						
Tukey HSD						
(I) 1.000000		Mean Difference (I- J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1 DALL-E	2	.917*	0.279	0.009	0.15	1.68
	3	-0.038	0.279	1.000	-0.80	0.72
	4	1.604*	0.279	0.000	0.84	2.37
	5	-0.110	0.279	0.995	-0.87	0.65
2 Stable Diffusio n	1	-.917*	0.279	0.009	-1.68	-0.15
	3	-.955*	0.278	0.006	-1.72	-0.19
	4	0.688	0.278	0.099	-0.07	1.45
	5	-1.027*	0.278	0.002	-1.79	-0.27
3 Imagen	1	0.038	0.279	1.000	-0.72	0.80
	2	.955*	0.278	0.006	0.19	1.72
	4	1.643*	0.278	0.000	0.88	2.40
	5	-0.071	0.278	0.999	-0.83	0.69
4 GROK	1	-1.604*	0.279	0.000	-2.37	-0.84
	2	-0.688	0.278	0.099	-1.45	0.07
	3	-1.643*	0.278	0.000	-2.40	-0.88
	5	-1.714*	0.278	0.000	-2.48	-0.95
5	1	0.110	0.279	0.995	-0.65	0.87

Real Image	2	1.027*	0.278	0.002	0.27	1.79
	3	0.071	0.278	0.999	-0.69	0.83
	4	1.714*	0.278	0.000	0.95	2.48
* The mean difference is significant at the 0.05 level.						

The highlighted values are less than 0.05, indicating that the group difference is significant in these cases. Considering this fact, the groups with differences between each other are illustrated as comparative graphs in figures 5.6 and 5.7.

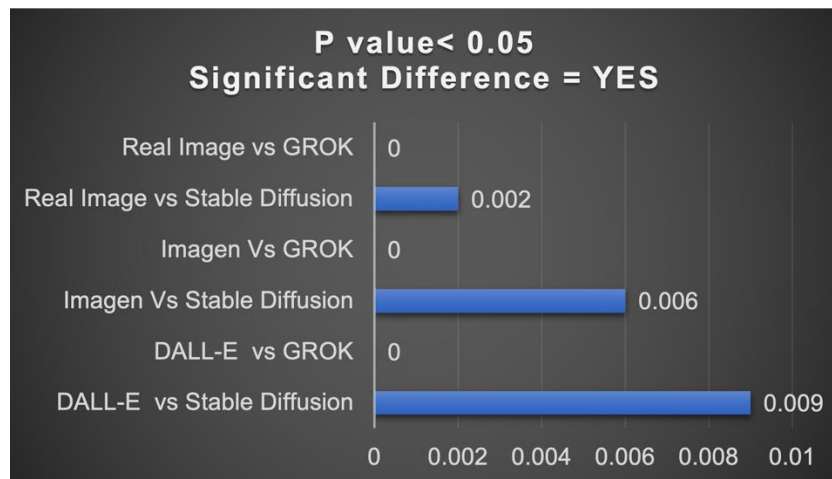


Figure 5. 6 Groups with Significant Difference

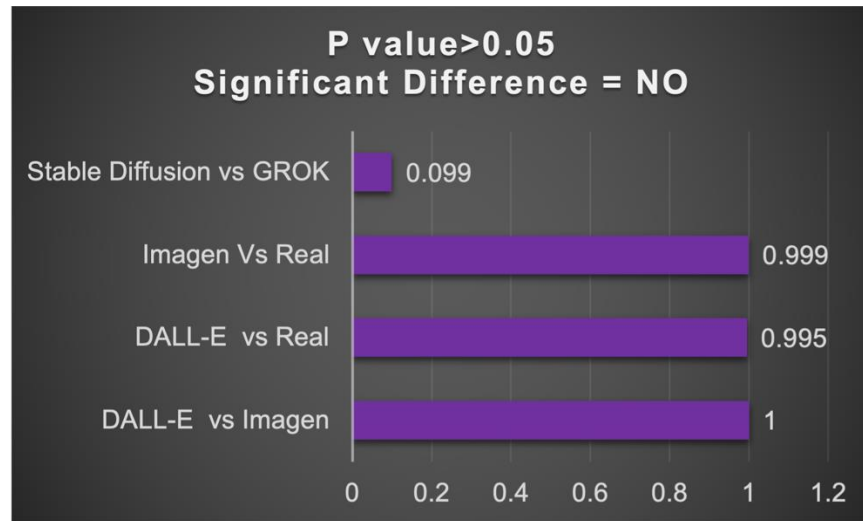


Figure 5. 7 Groups with No Significant Difference

5.5. DISCUSSION

In terms of mathematical evaluation, lower FID scores and higher SSIM and PSNR values generally indicate better image quality and similarity to real images. Referring to table 1, DALL-E exhibited the most promising performance, with the lowest FID score (9.0%) and highest SSIM (1.35%) and PSNR values (9.88), suggesting its capability to produce images that closely resemble real images. Conversely, Stable Diffusion demonstrated the highest FID score (15.95%) and lowest PSNR (9.21%) value, indicating potential limitations in generating realistic images compared to other models. These results provide valuable insights into the relative performance of these AI models in generating high-quality images.

Furthermore, the statistical analysis of the human perception survey data using Tukey's HSD test revealed significant differences in perceived realism between certain pairs of image-generative AI models. Like the mathematical evaluation, DALL-E showed significant differences in perceived realism compared to Stable Diffusion ($p = 0.009$), GROK ($p = 0$), and Real Image ($p = 0.002$), indicating that images generated by DALL-E were perceived as more realistic than those generated by these models. Similarly, Imagen did not exhibit significant differences in perceived realism compared to Real Image ($p = 0.999$), indicating that the perceived realism of images generated by Imagen was comparable to those generated by Real Image.

In summary, according to the participants' perceptions, the findings highlight the varying degrees of perceived realism among different image generative AI models, with DALL-E and Imagen generally being perceived as more realistic than Stable Diffusion and GROK. As examined, human evaluation is the current gold standard in text-to-image evaluation; however, mathematical based metrics also have promise and value. FID is growing as the standard evaluation method, and our results illustrate it most closely

represented human evaluation. However, this underscores the importance of considering both objective metrics and subjective human perception in evaluating the performance of image generative AI models.

5.6. CONCLUSION

The demand for AI-generated images continues to rise across various domains, understanding the capabilities and limitations of these models is crucial. By combining mathematical evaluation with human perception studies, we have comprehensively understood the relative performance of prominent text-to-image generative AI models. The mathematical evaluation of the image generative AI models indicates that lower FID scores and higher SSIM and PSNR values generally correspond to better image quality and similarity to real images. FID has emerged as a robust metric, often aligning closely with human perception as observed in our survey data. Therefore, FID could be considered a superior metric for mathematical evaluation compared to SSIM and PSNR. Additionally, the statistical analysis of the human perception survey data using Tukey's HSD test unveiled significant differences in perceived realism between certain pairs of image-generative AI models. DALL-E showed significant differences in perceived realism compared to Stable Diffusion ($p = 0.009$), GROK ($p = 0$), and Real Image ($p = 0.002$), suggesting that images generated by DALL-E were perceived as more realistic. Conversely, Imagen did not exhibit significant differences in perceived realism compared to Real Image ($p = 0.999$), indicating comparable perceived realism between images generated by Imagen and Real Image. The findings highlight varying degrees of perceived realism among different image-generative AI models, with DALL-E and Imagen generally perceived as more realistic than Stable Diffusion and GROK. This underscores the importance of considering both objective metrics and subjective human perception in evaluating the performance of image-generative AI models.

CHAPTER 6: CONCLUSION

Machine learning and artificial intelligence algorithms are increasingly vital in our lives due to their transformative impacts. These algorithms automate repetitive tasks, freeing up human resources for more complex and creative endeavors. They streamline processes in industries ranging from manufacturing and logistics to finance and healthcare, leading to increased efficiency and productivity [79-81]. Predictive modeling, leveraging ML and AI, facilitates the forecasting of future trends or behaviors based on historical data, while classification tasks play a crucial role in categorizing data into distinct classes, spanning from email filtering to medical diagnosis. Concurrently, the emergence of text-to-image generation represents a transformative potential, enabling the direct creation of visual content from textual descriptions. These advancements are significant in design, art, entertainment, and visual communication, fostering creativity and productivity. This paper has explored three significant studies in ML and AI research, focusing on predictive and classification solutions on cloud platforms. Firstly, a study evaluating regression-type ML models across cloud platforms provides crucial insights for optimizing model deployment strategies. Secondly, research on customizing large language models for email classification addresses cybersecurity concerns, thereby bolstering email security measures. Lastly, an investigation into text-to-image generation diffusion models highlights the evolving landscape of AI-driven visual content generation, informing future advancements and applications.

Finally, it can be said that these studies advance the applications of ML and AI technologies, addressing real-world challenges and driving innovation. The ongoing integration of ML and AI algorithms promises to unlock reshape industries, enhancing decision-making processes, and unlocking new possibilities for human-machine collaboration.

REFERENCES

- [1] D. M. Abdulqader, A. M. Abdulazeez, and D. Q. Zeebaree, "Machine learning supervised algorithms of gene selection: A review," *Machine Learning*, vol. 62, no. 03, pp. 233-244, 2020.
- [2] H. Meyer, M. Kühnlein, T. Appelhans, and T. Nauss, "Comparison of four machine learning algorithms for their applicability in satellite-based optical rainfall retrievals," *Atmospheric research*, vol. 169, pp. 424-433, 2016.
- [3] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128-138, 2017.
- [4] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140-147, 2020.
- [5] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," *Deep learning*, pp. 98-164, 2016.
- [6] V. V. Kolisetty and D. S. Rajput, "A review on the significance of machine learning for data analysis in big data," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 6, no. 01, pp. 155-171, 2020.
- [7] Y. Asim, A. R. Shahid, A. K. Malik, and B. Raza, "Significance of machine learning algorithms in professional blogger's classification," *Computers & Electrical Engineering*, vol. 65, pp. 461-473, 2018.
- [8] B. C. Love, "Comparing supervised and unsupervised category learning," *Psychonomic bulletin & review*, vol. 9, no. 4, pp. 829-835, 2002.
- [9] R. Ratra and P. Gulia, "Experimental evaluation of open source data mining tools (WEKA and Orange)," *International Journal of Engineering Trends and Technology*, vol. 68, no. 8, pp. 30-35, 2020.
- [10] S. Kodati and R. Vivekanandam, "Analysis of heart disease using in data mining tools Orange and Weka," *Global journal of computer science and technology*, vol. 18, no. 1, 2018.
- [11] S. Jamal, W. A. Elenin, and L. Chen, "Developing and Evaluating Data-Driven Heart Disease Prediction Models by Ensemble Methods on Different Data Mining Tools," in *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2023: IEEE, pp. 0678-0683.
- [12] S. Kavitha, S. Varuna, and R. Ramya, "A comparative analysis on linear regression and support vector regression," in *2016 online international conference on green engineering and technologies (IC-GET)*, 2016: IEEE, pp. 1-5.
- [13] S. Rajagopal, K. S. Hareesha, and P. P. Kundapur, "Performance analysis of binary and multiclass models using azure machine learning," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 10, no. 1, 2020.
- [14] P. Kaushik, A. M. Rao, D. P. Singh, S. Vashisht, and S. Gupta, "Cloud Computing and Comparison based on Service and Performance between Amazon AWS, Microsoft Azure, and Google Cloud," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 2021: IEEE, pp. 268-273.
- [15] A. Alkhatib, A. Al Sabbagh, and R. Maraqa, "Pubic Cloud Computing: Big Three Vendors," in *2021 International Conference on Information Technology (ICIT)*, 2021: IEEE, pp. 230-237.
- [16] D. Sikeridis, I. Papapanagiotou, B. P. Rimal, and M. Devetsikiotis, "A Comparative taxonomy and survey of public cloud infrastructure vendors," *arXiv preprint arXiv:1710.01476*, 2017.

- [17] N. Govil, K. Agarwal, A. Bansal, and A. Varshney, "A machine learning based spam detection mechanism," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020: IEEE, pp. 954-957.
- [18] C. Chen *et al.*, "A performance evaluation of machine learning-based streaming spam tweets detection," *IEEE Transactions on Computational social systems*, vol. 2, no. 3, pp. 65-76, 2015.
- [19] S. Kumar, X. Gao, I. Welch, and M. Mansoori, "A machine learning based web spam filtering approach," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016: IEEE, pp. 973-980.
- [20] H. Baaqeel and R. Zagrouba, "Hybrid SMS spam filtering system using machine learning techniques," in *2020 21st International Arab Conference on Information Technology (ACIT)*, 2020: IEEE, pp. 1-8.
- [21] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206-10222, 2009.
- [22] E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, 2019.
- [23] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in *Proceedings of the australasian computer science week multiconference*, 2017, pp. 1-8.
- [24] G. Chetty, H. Bui, and M. White, "Deep learning based spam detection system," in *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, 2019: IEEE, pp. 91-96.
- [25] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS performance evaluation review*, vol. 38, no. 1, pp. 367-368, 2010.
- [26] M. Manaa, A. Obaid, and M. Dosh, "Unsupervised approach for email spam filtering using data mining," *EAI Endorsed Transactions on Energy Web*, vol. 8, no. 36, 2021.
- [27] Y. Cabrera-León, P. García Báez, and C. P. Suárez-Araujo, "E-mail spam filter based on unsupervised neural architectures and thematic categories: design and analysis," in *International Joint Conference on Computational Intelligence*, 2016: Springer, pp. 239-262.
- [28] T. Jaya, R. Kanyaharini, and B. Navaneesh, "Appropriate Detection of HAM and Spam Emails Using Machine Learning Algorithm," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2023: IEEE, pp. 1-5.
- [29] A. Karim, S. Azam, B. Shanmugam, and K. Kannoopatti, "Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework," *IEEE Access*, vol. 8, pp. 154759-154788, 2020.
- [30] M. Ghiassi, S. Lee, and S. R. Gaikwad, "Sentiment analysis and spam filtering using the YAC2 clustering algorithm with transferability," *Computers & Industrial Engineering*, vol. 165, p. 107959, 2022.
- [31] Q. Yaseen, "Spam email detection using deep learning techniques," *Procedia Computer Science*, vol. 184, pp. 853-858, 2021.
- [32] X. Liu, H. Lu, and A. Nayak, "A spam transformer model for SMS spam detection," *IEEE Access*, vol. 9, pp. 80253-80263, 2021.
- [33] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Spam detection using bidirectional transformers and machine learning classifier algorithms," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 1, pp. 5-9, 2023.
- [34] V. S. Tida and S. Hsu, "Universal spam detection using transfer learning of BERT model," *arXiv preprint arXiv:2202.03480*, 2022.
- [35] I. Androutsopoulos, V. Metsis, and G. Paliouras, "The Enron-spam datasets," *Accessed: Oct*, vol. 11, p. 2019, 2006.
- [36] I. Androutsopoulos, "Ling-spam," *Aueb. gr. Accessed: Oct*, vol. 11, p. 2019, 2000.
- [37] Y. Wang, W. Zhu, H. Xu, Z. Qin, K. Ren, and W. Ma, "A Large-Scale Pretrained Deep Model for Phishing URL Detection," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023: IEEE, pp. 1-5.

- [38] P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, "URLTran: Improving phishing URL detection using transformers," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, 2021: IEEE, pp. 197-204.
- [39] K. Verma, T. Milosevic, K. Cortis, and B. Davis, "Benchmarking language models for cyberbullying identification and classification from social-media texts," in *Proceedings of the First Workshop on Language Technology and Resources for a Fair, Inclusive, and Safe Society within the 13th Language Resources and Evaluation Conference*, 2022, pp. 26-31.
- [40] K. Saifullah, M. I. Khan, S. Jamal, and I. H. Sarker, "Cyberbullying Text Identification based on Deep Learning and Transformer-based Language Models," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 11, no. 1, pp. e5-e5, 2024.
- [41] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," *arXiv preprint arXiv:1802.03162*, 2018.
- [42] F. Tajaddodianfar, J. W. Stokes, and A. Gururajan, "Texception: a character/word-level deep learning model for phishing URL detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020: IEEE, pp. 2857-2861.
- [43] X. Xu, Z. Wang, G. Zhang, K. Wang, and H. Shi, "Versatile diffusion: Text, images and variations all in one diffusion model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7754-7765.
- [44] A. Elarabawy, H. Kamath, and S. Denton, "Direct Inversion: Optimization-Free Text-Driven Real Image Editing with Diffusion Models," *arXiv e-prints*, p. arXiv: 2211.07825, 2022.
- [45] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22500-22510.
- [46] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18208-18218.
- [47] O. Avrahami, O. Fried, and D. Lischinski, "Blended latent diffusion," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1-11, 2023.
- [48] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12873-12883.
- [49] O. Gafni, A. Polyak, O. Ashual, S. Sheynin, D. Parikh, and Y. Taigman, "Make-a-scene: Scene-based text-to-image generation with human priors," in *European Conference on Computer Vision*, 2022: Springer, pp. 89-106.
- [50] R. Gal *et al.*, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618*, 2022.
- [51] M. Ding, W. Zheng, W. Hong, and J. Tang, "Cogview2: Faster and better text-to-image generation via hierarchical transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16890-16902, 2022.
- [52] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.
- [53] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684-10695.
- [54] M. Townsend, H. Wimmer, and J. Du, "Barriers and Drivers to Adoption of Cloud Infrastructure Services: A Security Perspective," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020: IEEE, pp. 1-7.
- [55] B. R. Parida, A. K. Rath, and H. Mohapatra, "Binary Self-Adaptive Salp Swarm optimization-Based dynamic Load Balancing in Cloud Computing," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 17, no. 1, pp. 1-25, 2022.

- [56] D. Ukene, H. Wimmer, and J. Kim, "Evaluating the Performance of Containerized Webservers against web servers on Virtual Machines using Bombardment and Siege," in *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, 2023: IEEE, pp. 144-152.
- [57] S. Jamal, M. V. Cruz, and J. Kim, "Cloud-Based Human Emotion Classification Model from EEG Signals," in *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2023: IEEE, pp. 0057-0064.
- [58] M. A. Rahman, J. Kim, F. Dababneh, and H. Taheri, "Railroad condition monitoring with distributed acoustic sensing: an investigation of deep learning methods for condition detection," *Journal of Applied Remote Sensing*, vol. 18, no. 1, pp. 016512-016512, 2024.
- [59] M. A. Rahman, H. Taheri, F. Dababneh, S. S. Karganroudi, and S. Arhamnamazi, "A review of distributed acoustic sensing applications for railroad condition monitoring," *Mechanical Systems and Signal Processing*, vol. 208, p. 110983, 2024.
- [60] M. A. Rahman, S. Jamal, and H. Taheri, "Remote Condition Monitoring of Rail tracks using Distributed Acoustic Sensing (DAS): A Deep CNN-LSTM-SW based Model," *Green Energy and Intelligent Transportation*, p. 100178, 2024.
- [61] S. Jamal and H. Wimmer, "An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach," *arXiv preprint arXiv:2311.04913*, 2023.
- [62] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)–Centric resource management in cloud computing: A review and future directions," *Journal of Network and Computer Applications*, p. 103405, 2022.
- [63] N. N. Grigoriou and A. Fink, "Cloud computing: Key to enabling smart production and industry 4.0," in *The Future of Smart Production for SMEs*: Springer, 2023, pp. 315-322.
- [64] F. B. Emdad, B. Ravuri, L. Ayinde, and M. I. Rahman, "" ChatGPT, a Friend or Foe for Education?" Analyzing the User's Perspectives on the Latest AI Chatbot Via Reddit," *arXiv preprint arXiv:2311.06264*, 2023.
- [65] L. Ayinde, M. P. Wibowo, B. Ravuri, and F. B. Emdad, "ChatGPT as an important tool in organizational management: A review of the literature," *Business Information Review*, vol. 40, no. 3, pp. 137-149, 2023.
- [66] J. Kim, J. Seol, T. A. Onisha, and Y. Ji, "Hyper Parameter Classification on Deep Learning Model for Cryptocurrency Price Prediction," in *2023 IEEE/ACIS 8th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 2023: IEEE, pp. 162-169.
- [67] S. Jamal and H. Wimmer, "Performance Analysis of Machine Learning Algorithm on Cloud Platforms: AWS vs Azure vs GCP," Cham, 2023: Springer Nature Switzerland, in *Information Technologies and Intelligent Decision Making Systems*, pp. 43-60.
- [68] J. Oppenlaender, "A taxonomy of prompt modifiers for text-to-image generation," *Behaviour & Information Technology*, pp. 1-14, 2023.
- [69] L. Yang *et al.*, "Diffusion models: A comprehensive survey of methods and applications," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1-39, 2023.
- [70] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 2249-2281, 2022.
- [71] Y. Zhu, Z. Li, T. Wang, M. He, and C. Yao, "Conditional Text Image Generation with Diffusion Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14235-14245.
- [72] G. Marcus, E. Davis, and S. Aaronson, "A very preliminary analysis of DALL-E 2," *arXiv preprint arXiv:2204.13807*, 2022.
- [73] S. Wang *et al.*, "Imagen editor and editbench: Advancing and evaluating text-guided image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18359-18369.

- [74] A. Obukhov and M. Krasnyanskiy, "Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance," in *Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020, Vol. I 4*, 2020: Springer, pp. 102-114.
- [75] M. Ponomarenko, K. Egiazarian, V. Lukin, and V. Abramova, "Structural similarity index with predictability of image blocks," in *2018 IEEE 17th International Conference on Mathematical Methods in Electromagnetic Theory (MMET)*, 2018: IEEE, pp. 115-118.
- [76] J. Kaufmann and A. Schering, "Analysis of variance ANOVA," *Wiley Encyclopedia of Clinical Trials*, 2007.
- [77] H. Abdi and L. J. Williams, "Tukey's honestly significant difference (HSD) test," *Encyclopedia of research design*, vol. 3, no. 1, pp. 1-5, 2010.
- [78] H. Zou, T. Hastie, and R. Tibshirani, "On the "degrees of freedom" of the lasso," 2007.
- [79] M. A. Rahman, H. Taheri, and J. Kim, "Deep Learning Model for Railroad Structural Health Monitoring via Distributed Acoustic Sensing," in *2023 26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter)*, 2023: IEEE, pp. 274-281.
- [80] F. B. Emdad, S. M. Ho, B. Ravuri, and S. Hussain, "Towards A Unified Utilitarian Ethics Framework for Healthcare Artificial Intelligence," 2023.
- [81] F. B. Emdad, S. Tian, E. Nandy, K. Hanna, and Z. He, "Towards Interpretable Multimodal Predictive Models for Early Mortality Prediction of Hemorrhagic Stroke Patients," *AMIA Summits on Translational Science Proceedings*, vol. 2023, p. 128, 2023.