Machines and Algorithms
http://www.knovell.org/mna

Editorial

Volume 4 Issue 2

Editor: Dr. Mahmood Ashraf
Department of Computer Science Times University, Multan, 60000, Pakistan

From the Editor

It is a great pleasure to present **Volume 4, Issue 2 (2025)** of the *Machines and Algorithms* journal. In this issue, we have presented various advanced researches relevant to the fields of machine intelligence, data sciences and computational models. These published papers validate the role of advanced machine learning algorithms are playing prominent role in advancing human understanding. The papers published in this issue belongs to several prominent domains, such that including quantum problem-solving, sentiment analysis, computer vision-based diseases diagnosis, automated AI based workflow analytics and environmental change forecasting. Collectively, all these researches are presenting the highly prominent role of AI and data science in industrial and scientific domains.

In the current issue, five multidimensional and thoroughly reviewed papers are being published, which are briefly introduced below:

The first paper is **"Sentiment Analysis on Tweets using N-Grams and Lexicon"**, which introduce a lexicon-based approach. The proposed approach refines textual noise in Twitter data and applies sentiment scoring from −1 to +1. Comparison of multiple n-gram configurations has been performed. Moreover, evaluation of sentiment accuracy against manually annotated benchmarks have also been performed. It provides insights into linguistic modeling for social media analytics.

The second contribution of this issue is, **"Systematic Literature Review on Problem Solving with Quantum Algorithms".** This paper presents a comprehensive synthesis of recent advances in quantum computation. Author of this study has examined thirty significant studies from 2015 to 2024. In this review, analysis of four cornerstone algorithms has been performed. These algorithms are: 1) Shor's, 2) Grover's, 3) QAOA, and 4) QFT. In addition to this, author of this study has also explored the applications of these algorithms in fields like optimization, cryptography, and machine learning. This work identifies significant and emerging research gaps of its field. Moreover, it also outlines promising future directions for inter-domain quantum applications.

The third article that has been published in this issue is, **"Deep Learning Architectures for Automated Ocular Disease Recognition".** This paper highlights the critical need of diagnosing ocular threatening diseases at an early stage i.e., diabetic retinopathy and glaucoma. Author of this study, has exploited transfer learning approach. More specifically, author has applied advanced CNN models like EfficientNet and InceptionResNetV2 for automated ocular disease diagnosis. The proposed approach has achieved a significant diagnostic accuracy of 98.2%. In addition to this, the presented method demonstrates how AI can make ocular screening more accessible for patients, especially in low-resource healthcare departments.

The fourth paper of this issue is, **"Robust Multi-Class Weather Classification from Images Using Deep CNN".** Author of this study, has proposed a computer-vision model for multi-class weather classification. The proposed model is capable of identifying five weather conditions including: clear, foggy, rainy, cloudy, and snowy, using weather imaging data. By achieving an accuracy of 85.2%, the proposed model also demonstrates its robustness in classification of different environmental conditions. Recommendations for enhancing similar future automated systems through transfer learning have also been provided in this study.

Finally, the last paper in this issue is, **"Predicting Employee Attrition Using XGB Classifier".** This paper investigates workforce analytics by the implication of optimized XGBoost classifier. For the training of this model HR dataset has been exploited. The proposed approach has achieved 87.76% prediction accuracy. Moreover, the model also highlights the potential significant variables including overtime, monthly income, and job satisfaction. Author of this study has also referred to the ethical considerations of transparency and fairness in predictive HR systems.

All of the above-mentioned papers reflect the dedication of our editorial board towards presenting highly and ethical AI based researches. I also want to extend my heartfelt appreciation to the authors for their precious scholarly efforts. I also appreciate to our reviewers for their thorough assessment of publications, and to the editorial team for ensuring the quality, excellence and integrity of published papers.

I am confident that all the readers of these papers, whether they are academic researchers or students, they will definitely find this issue both informative and inspiring. With the growth of our journal *Machines and Algorithms*, we are intended to expand the collaborations and introducing more thematic special issues. Furthermore, we are also intended to present more impactful and contributive papers for advancing computational innovation for the global research community.

Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Sentiment Analysis on Tweets using N-Grams and Lexicon

Muhammad Sanaullah [1, *], Rabea Saleem[2, *] and Fatima Riaz [3]

[1] Department of Computer Science, Bahauddin Zakariya University, Multan, 60800, Pakistan
[2] Department of Computer Science, Air University, Multan, 60000, Pakistan
[3] Lecturer Computer Science, Higher Education Department, Multan, 60000, Pakistan
[*]Corresponding Author: Muhammad Sanaullah. Email: drsanaullah@bzu.edu.pk

Received: 02 April 2025; Revised: 05 May 2025; Accepted: 04 July 2025; Published: 01 August 2025
AID: 004-02-000051

Abstract: Twitter is serving as micro-blogging platform where freedom is given to user to express their opinion and share information about any subject through short messages known as tweets. Tweet composed of textual data that can be classified into positive, negative or neutral sentiment. This classification is often based on the analysis of n-grams, which involves examining the frequency and combination of words used. This research article presents a technique which filters out the noise from tweets and apply the scoring mechanism to sentiments that assigns the score between -1 and +1. The proposed techniques results are validated by manually scored evaluations for same tweets. Additionally, the study compares the effectiveness of different n-gram techniques for sentiment analysis.

Keywords: Sentiment analysis; n-grams; twitter; tweets; lexicon;

1. Introduction

In this modern age of information, microblogging platforms and social media sites are the major source of information. These platforms allow user to express their feelings, emotions and their opinions. Various entities, including individuals, organizations, politicians, governments, and surveys, are increasingly utilizing these sites to fulfil their respective objectives. For instance, production companies may seek instant feedback on their products, political figures may engage in discussions on various issues, and governments may seek public opinions on policies and regulations.

In response to these objectives, a substantial volume of opinions is received from the respective users. Reviewing all of these opinions is a time-consuming and labor intensive task [1]. Consequently, there is a need for an automated tool that can classify these opinions, based on their nature, and allowing management to easily understand the behavior of their users. Sentiment Analysis, is a technique employed to analyze large amounts of data, serves the purpose of discerning user attitudes and opinions towards a given topic. Numerous machine learning techniques are utilized for opinion classification; however, due to inherent complexities in the text, such as unstructured content, negation detection, slang and abbreviation usage, irregular or incorrect wording, insufficient words, emoticons, symbols, tagging, and URLs, achieving the desired objective is not frequently attained [2].

To handle the inherent noise in such data and comprehend the underlying semantics of the text, an automated technique is essential. This technique should not only enable classification but also facilitate the measurement of the degree of emphasis in the text. To address this need, this paper proposes a comprehensive approach to sentiment analysis. Initially, the technique eliminates irrelevant information

and subsequently applies an n-grams (unigrams, bigrams, and trigrams) approach to analyze the sentiments expressed. The sentiment analysis is conducted utilizing SentiWordNet[3], a lexicon-based resource, to calculate the polarity score of the opinions, indicating the extent of positivity, negativity, or neutrality. These scores range from -1 (fully negative) to +1 (fully positive). These polarity scores are then employed to compute the sentiments and polarity of the opinions.

For the implementation of the proposed technique, Twitter was selected from among various microblogging sites, based on the following factors:

- The character limit of Twitter messages is restricted to 280 characters.
- Twitter has a larger user base consisting of more professional and engaged members.
- Many public figures utilize Twitter as a platform to discuss issues or policies.
- Twitter contains a significant volume of movie, event, and product reviews.

In the context of Twitter, user opinions are referred to as "tweets." To assess the reliability and accuracy of the proposed technique on these tweets, human experts were involved. These experts, unaware of the scores calculated by the technique, independently assigned similar scores to corresponding opinions, thereby confirming the technique's effectiveness.

The study contributes to present a novel technique for sentiment analysis on Twitter data, which filters out noise, assigns sentiment scores, and classifies tweets, with the reliability and accuracy of this approach validated through comparison with manually scored evaluations.

The content is the paper is structured as: the related work is presented in Section 2, the proposed technique is presented in Section 3, the implementation of the proposed technique is presented in Section 4, the results are shown in Section 5, and the last section consist of concluding remarks and future directions.

2. Related Work

In [4], authors gathered e-learning public opinions from twitter when the COVID-19 outbreak. People's opinions about the e-learning are classified into positive and negative polarities. The authors used the SVM model on the tweets and claimed that it performs even better than deep learning models. Most of the world agrees upon e-learning during the pandemic.

Another paper [5] worked on comparing the different methods of sentiment analysis on twitter gathered data. Authors also showed and compared the results. The proposed work showed the techniques of Bag of Words (BoW) and N-grams in a lexicon-based corpus. The work applied seven machine learning algorithms namely SVM, Naïve Bayes, Logistic Regression, Multi-layered perceptron, Best-first trees, functional trees and famous C4.5. Authors also tried combination of the algorithms and compared the results.

In another paper [6], the authors presented a hybrid approach for analysis of tweets. Authors searched the polarity of the words from Pool of Words in the lexicon-based corpus and trained the algorithms by polarity of the words.

In [7] article, the authors proposed a Naïve Bayes classifier that compares the 1-gram, 2-grams and 3-grams. The results showed that 2-grams work great and give best coverage of sentiments. Authors also used the negation attached with the words. This technique improved the classification process 2% on average. The authors distinguish the word occurrences that are ambiguous based on salience, entropy and sets. The two saliences showed good results but also created ambiguity in the system.

Another work [8] performed the sentiment analysis on microblogs and analyzed the role of word semantics. By focusing on semantics of the words give more accurate results and build good sentiment analysis models for twitter. Authors focus on two types of semantics: one based on context (also captured from words) and other concept-based semantics (gathered from other sources).

In [9], authors discussed a model based on semantics. This model identifies the tweet as an entity like Person, Organization, etc. Authors did not remove the stop words in their model because removing stop words effect the classifier. Most of the techniques used in sentiment rely on syntactic structure of words like great, bad. However, these techniques are considered weak because they do no tell the semantics of the

words in text. This paper improves the results by using basic sentiment analysis approaches in combination with stemming, two step classification and negation detection. Moreover, negation detection is also very important technique used in sentiment analysis to detect the negative words like ‘not’, ‘no’, ‘never, etc. By this the sentiments of the nearby words changed.

Paper [10] used the linguistic features of the language to detected the sentiment from tweets. Author captured the information about creative and informal language usually found in microblogging sites. For this, author utilized the supervised algorithms to the problem and trained the model on existing hashtags in the Twitter data.

In another paper [11], the authors worked on a lexicon-based sentiment discovering technique [8]. The lexicon used in the technique is built from the taxonomy. The taxonomy used in the sentiment technique contains positive, negative, negation, stop words and phrases. The text in the tweet usually contains the hashtags, emoticons, word variations, etc. Preprocessing of the tweet involves stemming, emoticon detection, hashtag detection, word shortening and normalization. The lexicon-based technique uses normalized tweet and classifies the tweet as negative, positive based on the contextual orientation of the words. The proposed system shows the F-score of 0.8004 when tested on the unseen tweets.

In [12], authors presented a model named LeBERT that used social media reviews and created an embedded model using BERT and sentiment lexicon. The authors used n-grams and also applied the CNN layers to the model and predicted a sentiment class. Author applied the model on three datasets about hotel, movies or products reviews. The f-score of 88.73% shows that the model outperforms most of the state of art models.

In this study [13], author introduced a novel approach for Sentiment Analysis with Convolutional Neural Network Optimized via Arithmetic Optimization Algorithm (TSA-CNN-AOA). The research extracted and analyzed 173,638 tweets from July 25, 2020, to August 30, 2020, utilizing FastText Skip-gram for information extraction. The convolutional neural network was used to extract feature and optimize feature selection through an arithmetic optimization algorithm (AOA). Classification of tweets as positive, negative, or neutral was performed using K-nearest neighbors (KNN), support vector machine, and decision tree algorithms. The results showed that TSA-CNN-AOA (KNN) achieved an impressive accuracy rate of 95.098%.

In this research [14], author proposed a hybrid approach combining lexicon-based methods with deep learning models to improve sentiment accuracy, assessing the impact of TextBlob compared to other methods like Afinn and VADER. Results show that models perform better with TextBlob-assigned sentiments. Proposed model stands out with a high accuracy of 0.97, and support vector and extra tree classifiers achieve top accuracy scores of 0.92, using TF-IDF and BoW.

In [15], authors presented a new framework that uses n-grams with existing lexicons. The framework is applied on three lexicon-based datasets. The authors experiments show that the proposed work outperforms the existing lexicon-based approaches. The proposed work utilizes the attribute vector containing binary value polarity scores without disturbing the word order of text instead of leveraging the average polarity score of the words in the sentence.

3. Methodology

The proposed methodology is based on six core steps, as shown in Figure 1, the work performed at each step is explained in the following subsections.

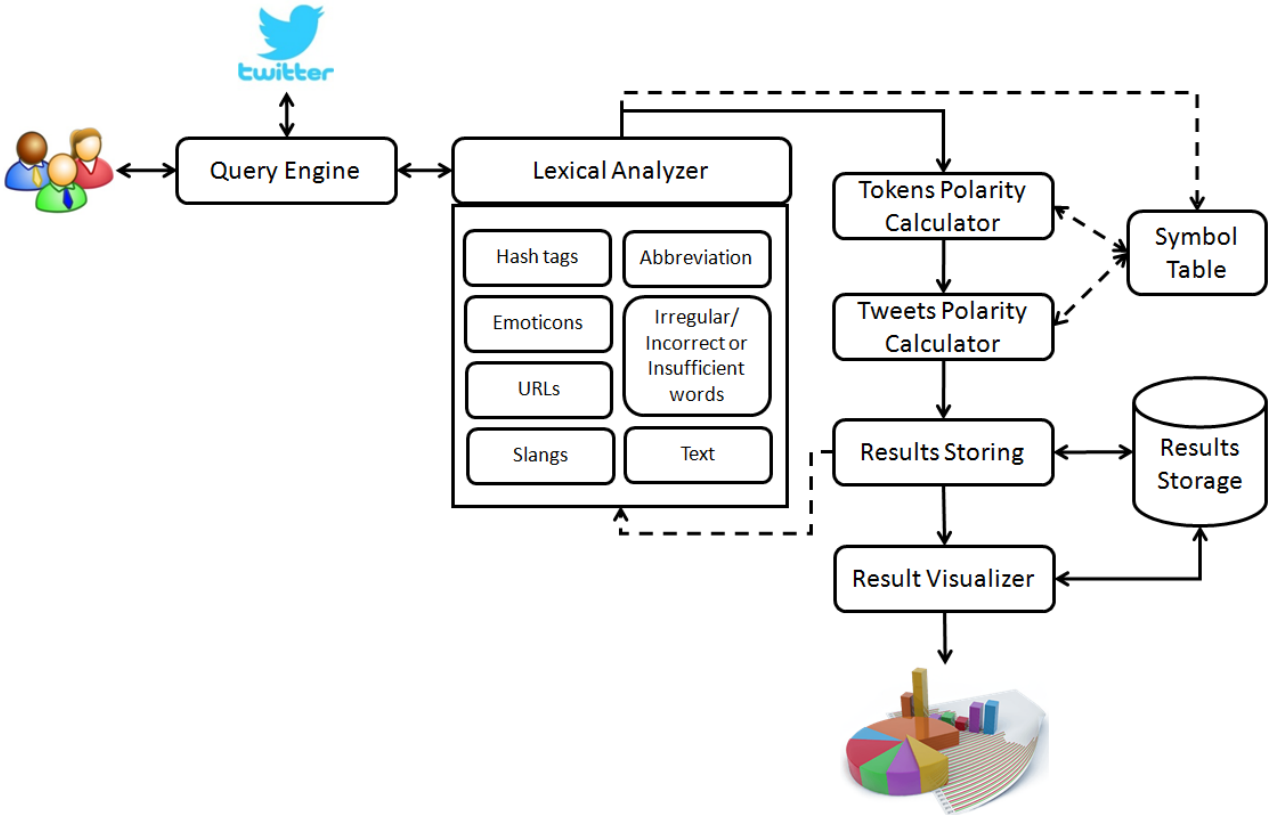


Figure 1: Methodology of Proposed Framework

3.1. Data Collection

In proposed system data set is collected from twitter in real-time. Users engage with the proposed system through a sophisticated interface. This interface enables users to submit a string or relevant keywords about the topic for which they seek sentiment analysis. For example, “Donald Trump”. The submitted keyword, denoted as Q (Quey), is then transmitted to the Query Engine (QE). The QE leverages the Twitter API to extract a set of real-time tweets, denoted as T, from twitter. Total number of tweets extracted from twitter is 100. The extraction time for tweets is a few seconds. The mathematical representation of the process can be expressed by the following equations.

- Q = Query (Keyword that is used to search tweets.)
- T = Extracted Tweets based on keyword.
- Q➔T
- T= [T1, T2, T3, … T_N]

The data extraction process relies on the utilization of the Twitter API. This API facilitates the collection of data, specifically tweets, from twitter platform. By creating a Twitter API, users gain access to and can store the desired data. Users initiate requests to the API to obtain from twitter data, and the API responds by returning data that aligns with the user's query.

3.2. Lexical Analyzer

After getting tweets from Twitter API, preprocessing is performed on tweets to make them more refined. Predefined python libraries Tweepy, NumPy, NLTK, pandas, csv, re, matplotlib and TextBlob are used in preprocessing. Preprocessing tasks involved in Lexical Analyzer are given below.

1. **Tokenization:** The process of splitting the input text into individual words is tokenization. For example, the sentence "I love this Car" would be tokenized into three tokens: ["I", "love", "this”,

"Car"]. Proposed system performs n-grams (unigram, bigram, trigram) tokenization one by one to perform more accurate sentiment analysis

2. **Stop Words Removal:** The words that do not has useful meanings and used as supporting words in sentences, known as stop words (e.g., "the," "and," "in"). These stop words increase the noise in our text. So, we have to remove these words.
3. **Lowercasing:** The process of conversion of capital letters to small letters is known as Lowercasing. In computer capital letter is different from lower case letter. The letter "M" is different from "m" in computer understanding. That's why the text is converted into lowercase to remove the uncertainty.
4. **Punctuation and Special Characters:** Punctuation marks and special characters can have sentiments. For example, "I love this!" and "I love this." might have different sentiment due to the presence of exclamation marks and periods.
5. **Lemmatization:** The process of reducing words to its root form is known as Lemmatization. Lemmatization is applied to capture the accurate semantics of a word. For instance, "running" and "ran" may be reduced to "run."

Our Lexical analyzer performs the tasks that are given below.

1. **URLs:** Twitter users engage in the platform not only to express their views but also to disseminate valuable information among others. One common method of sharing information involves the inclusion of links within tweets. These links, often in the form of URLs such as http://plurk.com/p/116r50, do not contribute to determining the sentiment of the tweet. Consequently, during the pre-processing stage, these URLs are removed to ensure they do not influence the sentiment analysis process.
2. **Usernames:** Within tweets, the "@" symbol is employed as a reference to mention or direct a tweet to other individuals. However, these references do not contribute to the sentiment analysis process. Hence, as part of the pre-processing stage, these references are removed to ensure they do not affect the sentiment analysis.
3. **Duplicates or repeated characters:** Twitter users often employ casual language and may use words in altered forms. For instance, the word "happy" might appear as "haaaaaaappy." Despite the variation in spelling, the underlying sentiment remains the same. To address this issue, the pre-processing stage involves the removal of duplicates and repeated words. These instances are then replaced with the correct form of the word to ensure accurate sentiment analysis, as shown in Table 1.
4. **Emotions:** Within tweets, various emoticons and emotion symbols such as 😊 and ☹ are frequently utilized. However, these symbols do not directly contribute to sentiment analysis. As a result, during the pre-processing stage, they are replaced with corresponding sentiment labels to ensure their influence on sentiment determination is accounted for. For example, "😊" replaced with "Happy," while "☹" replaced with "Sad." This substitution allows for a more accurate assessment of sentiment within the tweet content, as shown in Table 1.
5. **Stop-words removal:** In the context of information retrieval, numerous words serve as conjunctions within sentences. The stop-words examples include "the", "and", "before" and "while" etc. These conjunctive words don't have the significant impact on the sentiments of the tweets. Also, the stop-words don't help in the classification of all the classes of tweets.
6. **Spellchecker:** In case of misspelled words found in tweets. To address this issue pre-processing stage involves the correction of spelling for a more accurate assessment of sentiment within the tweets content.

3.3. Use of Dictionaries

Preprocessing procedures also make use of dictionaries in text documents.

1. Slang.txt: Used to detect and replace slangs.
2. Affin.txt: Used to detect and replace emotion with word.

3. Wordlist.txt: Used to detect hashtags and arrange in sequence with words space.

Table 1: Tweets preprocessing

Tweets Containing Words	Replaced By
Coooooooooool	Cool
Baaaaaad	Bad
😊	Happy
☹	Sad
#ILovePakistan	I love Pakistan
Lol	Lot of laughs
Apparent	Apparent

3.4. Tokens Priority Calculator

Tweets are tokenized, converted into n-grams including unigrams, bigrams, and trigrams. Words have the capacity to possess the different meanings according to their position in the sentence. In certain cases, combination of words offers more precise meanings as compared to individual words meaning. By using n-gram deeper understanding of sentence can be achieved. N-grams has numerous applications in text mining and natural language processing tasks, enabling more comprehensive analysis and interpretation of textual data.

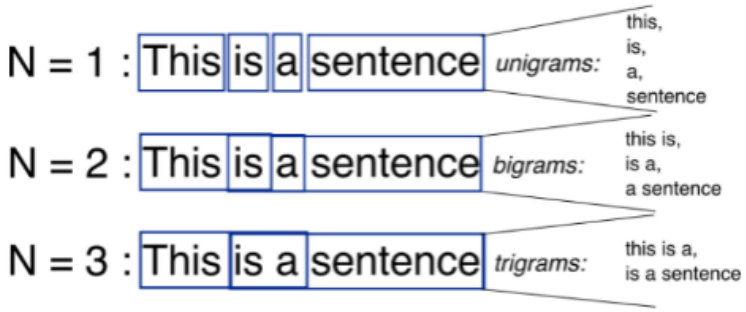


Figure 2: n-grams

The classification accuracy of tweets can be increased using n-grams as compare to single word feature. For the lexical analysis of the Tweets the three approaches can be adopted:

3.4.1. Manual Approach

The sentiment of words is based on the language understanding of a specific domain. This manual approach needs a human intervention that can be time-consuming. Typically, this approach is combined with automated approaches to streamline the sentiment analysis process. By combining manual and automated techniques, a more efficient and effective sentiment analysis methodology can be achieved.

3.4.2. Dictionary-Based Approach

The proposed technique of this paper uses the online dictionaries to determine the orientation of words. This approach does not require any specific domain knowledge in order to work. These dictionaries employ various techniques, such as synonyms, antonyms, and word hierarchies, to determine the sentiment of the word. However, context-specific sentiment can be challenging to identify using only a dictionary-based

approach. Popular online dictionaries utilized in this approach include WordNet, SentiWordNet, secticNet, and SentiFul, among others. These dictionaries serve as valuable resources in the process of sentiment analysis for determining the sentiment polarity of words.

3.4.3. Corpus-Based Approach

This approach takes into consideration the syntactic patterns and co-occurrence of words in the text along with their sentiment. This approach addresses some limitations of the dictionary-based approach. It relies on labelled data for training and analysis purposes. However, both the corpus-based approach and the manual approach are generally considered to be less efficient as compared to the dictionary-based approach. Nevertheless, the corpus-based approach offers more advantage of capturing contextual nuances and patterns that may not be readily apparent in dictionary-based methods.

4. SentiWordNet

Different tools TextBlob, Vader and SentiWordNet are used to perform lexicon-based analysis. In proposed work for n-gram analysis SentiWordNet is a lexical resource specifically designed to assist in Sentiment Analysis applications. It offers annotations in the form of three numerical sentiment scores (positivity, negativity, neutrality) for each synset within WordNet. These sentiment scores are the valuable indicators of the sentiment polarity associated with the corresponding synsets. This approach facilitates in getting the more accurate sentiment analysis of textual data. It has vast vocabulary that is easily adaptable to a variety of domain SentiWordnet's lexical relation makes n-gram analysis easier to handle. Words are mapped to synsets using Pos Tagging Integration, which increased the accuracy of the study.

Conversely, however TextBlob and Vader are two more lexicon-based programs that offer a single polarity score. Their vocabulary are limited, and they are unable to apply it for a variety of topics. These tools do not offer deeper analysis for n-grams since Vader is used for social media bias and produces good results for brief expressions, while TextBlob is the most basic tool that treats words in isolation.

In proposed work after converting the tweet to n-gram, the lexicon-based resource SentiWordNet is employed to determine the scores of the n-gram tokens. By using SentiWordNet, the sentiment scores n-gram tokens can be obtained that allowing for a more comprehensive analysis of the sentiment tweets.

5. Tweet Polarity Calculator

The tweet polarity is computed by using scores of unigrams, bigrams, and trigrams for each tweet. Additionally, assigned weights are applied to tweet objects such as favcount, likes, and hashtags.

The tweet polarity of each n-gram group is determined using the following methodology:

5.1. Unigrams

In Unigrams, the tweet polarity is calculated by adding the SentiWordNet scores for the unigram words. SentiWordNet gives us the (pos, neg) values for each phrase score. We must determine whether the sentence total emotion score is positive or negative. We utilized equation_1 and equation_2 to determine the *Average UniScore*. A CSV (Comma Separated Value) file is created by calculating and storing the average score. This makes it possible to store the sentence polarity scores of each unique unigram in an organized manner and analyze them further.

- Pos = positive value
- Neg = negative value
- Uni = unigrams
- Tokens = unigrams words
- Equation_1. Calculate the total score of sentences for both positive and negative.
- Equation_2. Used to determine the average score of unigram words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.

$$Uni_{Sum(pos,neg)} = \sum_{n=1}^{n} Word_1\,(pos, neg) + Word_2(pos, neg) + \cdots + Word_n(pos, neg) \quad (1)$$

$$Average\ UniScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Uni\_\mathrm{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (2)$$

5.2. Bigrams

In Bigrams, the tweet text is transformed into groups of two co-occurring words. Consequently, the sentence polarity is computed by taking the product of the scores of the two words in each bigram. The scores of all the bigram groups with multiplied word scores are then summed. After summing the scores, the average of the resulting sum is calculated. The computed average value is then stored in a CSV file. This facilitates organized storage and subsequent analysis of the tweet polarity scores associated with the bigrams.

- Equation_3. Multiply two-word groups that occur together, then execute the summation to determine the overall score of sentences for both positive and negative.
- Equation_4. Used to determine the average score of bigrams words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.
- Tokens = Bigrams words
- Bi = Bigrams

$$Bi_{Sum(pos,neg)} = \sum_{n=1}^{n}\big(Word_1(pos, neg) * Word_2(pos, neg)\big) + \cdots + (Word_n(pos, neg) * Word_{n-1}(pos, neg)) \quad (3)$$

$$Average\ BiScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Bi\_\mathrm{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (4)$$

5.3. Trigrams

In the case of **Trigrams**, the tweet text is converted into groups of three co-occurring words. The sentence polarity in trigrams is determined by taking the product of the scores of the three words within each trigram. The scores of all the trigram groups with multiplied word scores are then summed. Once the scores are added together, the average of the resulting sum is calculated. This average value is then stored in a CSV (Comma Separated Value) file format, allowing for organized storage and subsequent analysis of the tweet polarity scores associated with the trigrams.

- Equation_5. Multiply three-word groups that occur together, then execute the summation to determine the overall score of sentences for both positive and negative.
- Equation_6. Used to determine the average score of trigrams words on a positive-negative scale; sentence polarity is positive if the average score value is greater than 0 or negative otherwise.
- Tokens = trigrams words
- Tri = Trigrams

$$Tri\_Sum(pos, neg) = \sum_{n=1}^{n}\big(Word_1(pos, neg) * Word_2(pos, neg) * Word_3(pos, neg)\big) + \cdots + (Word_n(pos, neg) * Word_{n-2}(pos, neg) * Word_{n-1}(pos, neg)) \quad (5)$$

$$Average\ BiScore = \begin{cases} \frac{\sum_{n=1}^{n} Word_n(pos)}{Total\ no.\ of\ tokens}, & if\ Tri\_\text{Sum}(pos) > 0 \\ \frac{\sum_{n=1}^{n} Word_n(neg)}{Total\ no.\ of\ tokens}, & otherwise \end{cases} \quad (6)$$

After calculating tweet polarity by using the above formulas stored them in a CSV file and showed the results in a graph and tabular form.

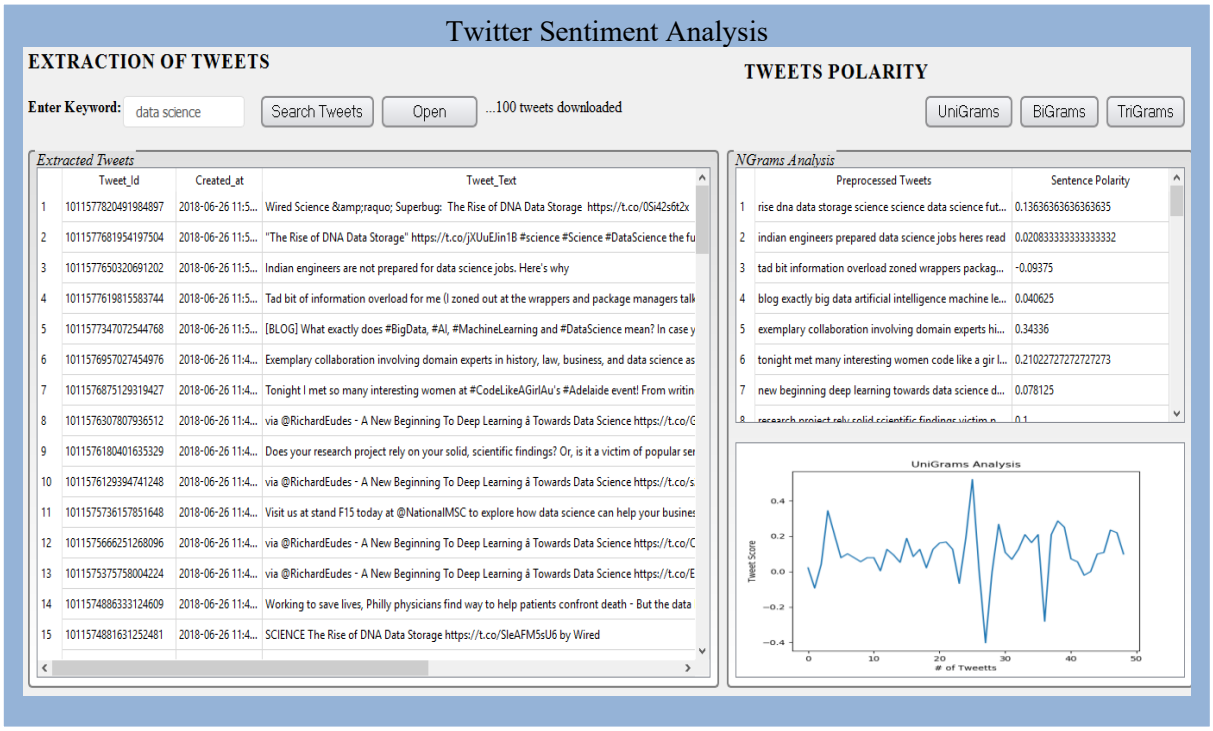


Figure 3: Interface of Proposed Work

6. Implementation

The proposed system requires the proper download and installation of the following components:

1. **Python 2.6 or above:** Python programming language should be installed and configured correctly in the desired location.
2. **Tweepy library:** This library is used to gather the tweets from Twitter by using Twitter API. It is a wrapper that provides easy retrieving and managing of the Twitter data [11].
3. **NumPy:** This library provides easy operations on multi-dimensional arrays and large mathematical functions [11].
4. **NLTK (Natural Language Toolkit):** This toolkit provides wide variety of tools to work with textual data. It also provides all the text processing libraires like classification, tokenization, stemming, parsing and semantic reasoning and Dictionary like WordNet.[11]

6.1. Pandas library

This library provides fast and flexible management of large labelled or unlabeled data in the form of a Dataframe. It is a versatile library for data analysis and manipulation of real-world data tasks.

1. **CSV (Comma Separated Value) library:** This library helps in reading and writing the tabular data in csv files.
2. **RE:** This module provides the support for regular expression matching operations. It allows for the matching of patterns against Unicode strings and 8-bit strings.
3. **Extract Data:** Tweepy is the open-source library that is used to extract the tweets from twitter.
4. **Twitter Posts:** The tweets are extracted and stored in CSV file.

6.2. Pre-processing

The extracted tweets are initially in an unstructured form, referred to as noisy data, which contains irrelevant information that cannot be used in the Sentiment Analysis (SA) process. Therefore, pre-processing of the tweets is performed to clean and refine the data by removing unwanted elements such as common words, stop words, symbols, extra spaces, special characters, and numbers. Additionally, the uppercase letters are converted to lowercase to ensure consistency in the data.

In the proposed system, the Python standard library is utilized for performing the pre-processing steps. Some of the key steps involved are as follows:

- Converting uppercase letters to lowercase: This step ensures uniformity in the text data, treating uppercase and lowercase letters as the same.
- Filtering URLs: URLs can be filtered using regular expressions such as [a-zA-Z0-9\./] +, which matches the URL pattern. The URLs are then replaced with the term "URL" to eliminate their influence on sentiment analysis.
- Removing user references (@): User references in the form of "@username" can be removed using regular expressions, specifically @(\w+), which matches the "@" symbol followed by one or more-word characters.
- Removing hashtags (#): Hashtags, denoted by the "#" symbol, can be removed using regular expressions, such as #(\w+), which matches the "#" symbol followed by one or more-word characters.
- Removing repeated characters: In colloquial language, words with repeated characters, such as "I'm in a hurrryyyyy," can be cleaned by using regular expressions. For example, (.)\1Error! Bookmark not defined. matches any character followed by one or more repetitions of the same character, and \1\1 is used to replace the repeating characters with a single instance.

The pro-processing steps are performed to clean the tweets data and used this clean data for further analysis.

6.3. n-grams

In the proposed system, three approaches of n-gram (unigrams, bigrams, and trigrams) are used for experimentation. Each tweet text is converted into these n-gram groups to capture different levels of linguistic context.

- Unigrams: In this step, the tweet text is tokenized into individual words and forming the unigrams. Each word in the tweet represents a single unigram. This approach allows us to analyse the sentiment associated with each word in the tweet.
- Bigrams: The tweet text is further processed to identify co-occurring pairs of words, known as bigrams. By grouping words in pairs, we can capture more contextual information and understand the sentiment that arises from word combinations. For example, "good movie" or "happy birthday" would be considered as bigrams.
- Trigrams: To get deeper into the linguistic context, the tweet text is transformed into trigrams,

which consist of three consecutive words. Trigrams helps to capture more complex relationships between words in a sentence and provide a higher level of context for sentiment analysis. Examples of trigrams could include "sky is blue" or "great customer service."

After converting the tweet text into n-gram, the proposed system gets different levels of linguistic information to better analyze the sentiment expressed in the tweets. Each type of n-gram allows for a more understanding of the text, leading to improved sentiment classification and analysis results.

6.4. Tweet Polarity

After converting the tweet text into n-gram, the score is assigned to each word by using the SentiWordNet dictionary. After scoring proposed system calculates the polarity using the SentiWordNet dictionary. The polarity of the tweet depends on both positive and negative scores. The positive and negative scores are added up for all the n-grams tokens in the tweet. The final polarity score represents sentiment expressed in the tweet.

The final polarity score has the range of [-1, 1]. The polarity score after zero (inclusive) is considered positive. Conversely, if the score is negative, it is classified as negative sentiment. For example, tweet: "I absolutely loved the movie! The acting was phenomenal."

6.4.1. Unigrams

Tokenized unigrams: ["i", "absolutely", "loved", "the", "movie", "The", "acting", "was", "phenomenal"]

The positive and negative scores of each unigram are obtained from the SentiWordNet dictionary after applying the equation_1and equation_2.

6.4.2. Bigrams

Tokenized bigrams: ["i absolutely", "absolutely loved", "loved the", "the movie", "movie acting", "acting was", "was phenomenal"]. The positive and negative scores of each bigram are obtained from the SentiWordNet dictionary after applying the equation_3 and equation_4.

6.4.3. Trigrams

Tokenized trigrams: ["i absolutely loved", "absolutely loved the", "loved the movie", "the movie acting", “movie acting was”, "acting was phenomenal"]

The positive and negative scores of each trigram are obtained from the SentiWordNet dictionary after applying the equation_5 and equation_6.

6.5. Unigrams Scoring

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com.
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Unigrams Tokenization:** “i”, “like”, “foxy”, “car”, “happy”
- **Calculated Polarity:** I: positive :0.01 negative:0.0, like: positive :0.2 negative: 0.01, foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01, happy: positive:0.125 negative:0.01
- Unigrams Sentence Polarity=0.24
- **Added Assigned Weight:** In the proposed system, the polarity calculation for unigrams takes into account the occurrence of emotions, the tweet length, and the total number of likes. Emotions are weighted and added to the polarity score, the tweet length is multiplied by its weight and added, and the total number of likes is also incorporated. Hashtags, however, are not considered in the polarity calculation.

6.6. Bigrams Scoring

In the case of bigrams, the tweet text is divided into pairs of words. The polarity score is obtained for each word using SentiWordNet (SWN). Then, the scores of each word pair are multiplied together and added up to calculate the sentence polarity. If the final score is greater than '0', it is considered positive; otherwise, it is considered negative.

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Bigrams Tokenization:** “i like", "like foxy", "foxy car", "car happy"
- **Calculated Polarity:** 1st step (i: positive :0.01 negative:0.0, like: positive:0.2 negative: 0.01), (like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0) (foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01) (car: positive:0.1 negative:0.01 happy: positive:0.125 negative:0.01)
- 2nd Step Multiply the positive score with a positive and negative score with a negative in every bigram group. It shows the relation between two co-occurring words.
- (I like: positive :0.03, negative:0.0), (like foxy: positive:0.05, negative: 0.0) (foxy car: positive:0.025, negative:0.0) (car happy: positive:0.0125, negative:0.0001)
- Add all of them and take the average for sentence polarity.
- Bigrams Sentence Polarity=0.145
- **Added Assigned Weight:** Emotion count, tweet length, and total number of likes are taken into account. These counts are multiplied by their respective weights and added to the polarity score of bigrams. If there are no hashtags in the example, the hashtag count is considered as '0'.

6.7. Trigrams Scoring

In the case of trigrams, there are three groups of words. The polarity calculation involves multiplying the scores of each word group obtained from SentiWordNet and then adding them together to determine the sentence polarity. If the final score is greater than '0', it is considered positive; otherwise, it is considered negative.

- **Tweet:** I like Foxy car 😊….!!!! https://www.pakwheels.com.
- **Preprocessed:** detected emotion symbol and convert into text. Removed stop words and URL.
- **Trigrams Tokenization:** “i like foxy", " like foxy car", "foxy car happy"
- **Calculated Polarity:** Step 1 (i: positive :0.01 negative:0.0, like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0), (like: positive:0.2 negative: 0.01, foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01) (foxy: positive:0.25 negative:0.0, car: positive:0.1 negative:0.01 happy: positive:0.125 negative:0.01)
- **Step 2:** Multiply the positive score with the positive and negative score with the negative in every trigram group. It shows the relation of three co-occurring words.
  (i like foxy: pos :0.03, neg:0.0), (like foxy car: pos:0.05, neg: 0.0) (foxy car happy: pos:0.025, neg:0.0)
- After adding a score, calculated the average of tweet polarity.
- Trigrams Tweet Polarity=0.002
- **Added Assigned Weight:** The proposed system uses emotions, tweet’s length, and the number of likes to compute the polarity score. The system counts the number of emotions in each tweet, multiplies it by the assigned weight, and adds it to the trigrams polarity score. Similarly, tweet’s length and the number of likes is multiplied by their respective weights and added to the trigram’s polarity score. In the given example, since there are no hashtags, their count is considered as '0'.

7. Results

7.1. Survey Results

To evaluate the efficiency of the proposed work, a survey was conducted on Twitter datasets. Dataset was consisting on 50 tweets. A random tweets list was generated, no. of 100 graduate college students and teachers' opinion are taken regarding the tweets were positive, negative or neutral. Tweets were presented them in printed document. Performa was design that was consist on 3 columns. *Sr no.*, *Tweet*, *Sentiment Score* (Pos, Neg, Neu). Participant tick on one option (pos, neg, neu) after reading the tweets. After getting result we performed n-gram (unigram, bigram, trigram) analysis of those tweets in proposed system and compared it with survey result.

Table 2: Survey Results

Tweets	Peoples Opinion (Survey Result)	Unigrams	Bigrams	Trigrams
Tweet_1	Pos (75%), Neg (0%), Neu (25%)	Pos	Pos	Pos
Tweet_2	Pos (60%), Neg (0%), Neu (40%)	Neg	Pos	Neu
Tweet_3	Pos (40%), Neg (50%), Neu (10%)	Neg	Neg	Neu
Tweet_4	Pos (70%), Neg (5%), Neu (25%)	Pos	Pos	Pos
…				
Tweet_50	Pos (70%), Neg (5%), Neu (25%)	Pos	Pos	Pos

7.2. Comparison of Results

From the survey results, it was found that 75% of the participants expressed a positive opinion about tweet_1, while 25% considered it to be neutral. None of the participants had a negative opinion about the tweet. In the proposed work, the sentiment analysis results for tweet_1 were as follows: the unigrams analysis indicated a positive sentiment, while both the bigrams and trigrams analyses yielded neutral sentiments. In rest of tweets 41% results match with unigram, 44% results match with bigrams and 36% results match with trigrams out of 50 tweets.

7.3. Accuracy of Results

From the survey results, the accuracy of the proposed work is calculated by using an Accuracy formula.

$$Accuracy = Total\ no.of\ correct\ queries / Total\ no.of\ queries \quad (7)$$

Unigrams = 80% Bigrams = 88% Trigrams= 72%

Comparison of survey results with proposed system bigrams > unigrams and trigrams. It means bigrams produce more accurate result.

8. Discussion

The sentiment analysis results using unigrams (single words), bigrams (pairs of words), and trigrams (three words) show some interesting findings about how well these different methods work. The results suggest that bigrams are more effective than unigrams and trigrams. This is because bigrams look at pairs

of words together, which helps to better understand the context and handle things like negations. By considering two words at a time, the meaning of the sentence is clearer, leading to more accurate sentiment analysis.

The trigram analysis, which looks at groups of three words, didn't show clear results because all the sentiment scores were '0'. This suggests that using three-word combinations might not be as helpful in this case. Trigrams add more complexity but don't improve sentiment analysis much in this context.

From these results, we can conclude that both unigrams (single words) and bigrams (pairs of words) are good choices for analyzing opinions in text. Using both methods together is especially effective for understanding sentiments, as it improves the accuracy and efficiency of the analysis in the system being studied.

To check efficiency different twitter datasets were used and found good results.

Table 3: Comparison with different twitter datasets.

Ref	Technique	Analysis Feature	Accuracy
[7]	Supervised and Unsupervised Learning	Evaluate the performance of existing technique	ML=86.40 DeepLearning=80.70 LexiconBased=74.00
[8]	Machine Learning	Unigrams, Bigrams, Trigrams	Bigrams>Unigrams and Trigrams
[16]	Lexicon Based and Machine Learning	Bag of Words, N-grams	83.15 on n-grams with combinatory approach
[17]	Hybrid	Unigrams, Bigrams, Trigrams	Unigrams = 63.23 Bigrams= 62.33 Trigrams=59.98
[18]	Lexicon Based	Stemming, Emoticon Detection and Normalization, Exaggerated word shortening and Hashtags Detection	0.800 F-Score
Proposed System	Lexicon Based (Used SentiWordNet)	Emotion Detection, HT's Detection, Slangs Detection Spelling Correction Uni, Bi, Tri	Unigrams= 80 Bigrams= 88 Trigrams= 72

9. Future work

Sentiment Analysis on Twitter data is very tricky task. In our work, we discovered that how to understand the semantic of meaning and improve the accuracy of polarity on unlabeled data. Preprocessing phase is playing very vital role in existing work because, if preprocessing done good then calculation of polarity will give good results.

There are many factors that can be participate for future work like detection of Sarcasm, ruled based negation, extraction of useful video links and stemming can be improve the sentiment results more refine and useful. However, SentiWordNet has a limited ability to identify sarcasm. In the future, we can combine Vadar and SentiWordNet in a hybrid mode to deal with sarcasm and rule-based negation.

Author Contributions

Muhammad Sanaullah examined and proposed the experiments, conducted it, designed proposed work interface, evaluated the data, designed the mathematical computations, formatted the figures and/or tables, composed or revised article drafts and approved the final draft.

Rabea Saleem examined and proposed the experiments, conducted it, performed the mathematical computations, developed interface of proposed work, composed or revised article drafts.

Fatima Riaz examined and proposed the experiments, conducted it, evaluated the data, evaluated the mathematical computations, performed testing, formatted the figures and/or tables, composed or revised article drafts.

Funding Statement: The authors received no funding to conduct this study.

Conflicts of Interest: Authors declare that no conflicts of interest exist regarding this study.

Data Availability: The data supporting this study were collected in real-time from Twitter using the Twitter API and are available upon reasonable request, subject to Twitter's data sharing policies.

References

[1] B. Shelke, Mahesh, Daivat D. Sawant, Chatrabhuj B. Kadam, Kailas Ambhure, and Sachin N. Deshmukh. "Marathi SentiWordNet: A lexical resource for sentiment analysis of Marathi." *Concurrency and Computation: Practice and Experience* 35, no. 2 (2023): e7497.

[2] Kaur, Ravneet, Ayush Majumdar, Priya Sharma, and Bhavana Tiple. "Analysis of tweets with emoticons for sentiment detection using classification techniques." In *International Conference on Distributed Computing and Intelligent Technology*, pp. 208-223. Cham: Springer Nature Switzerland, 2023.

[3] Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In *Lrec*, vol. 10, no. 2010, pp. 2200-2204. 2010.

[4] Kristiyanti, Dinar Ajeng, Dwi Andini Putri, Elly Indrayuni, Acmad Nurhadi, and Akhmad Hairul Umam. "Twitter sentiment analysis using support vector machine and deep learning model in e-learning implementation during the Covid-19 outbreak." In *2ND INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SCIENTIFIC DEVELOPMENT (ICAISD) 2021: Innovating Scientific Learning for Deep Communication*, vol. 2714, no. 1, p. 020033. AIP Publishing LLC, 2023.

[5] Psomakelis, Evangelos, Konstantinos Tserpes, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. "Comparing methods for twitter sentiment analysis." *arXiv preprint arXiv:1505.02973* (2015).

[6] Lalji, T., and Sachin Deshmukh. "Twitter sentiment analysis using hybrid approach." *International Research Journal of Engineering and Technology* 3, no. 6 (2016): 2887-2890.

[7] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In *LREc*, vol. 10, no. 2010, pp. 1320-1326. 2010.

[8] Saif, Hassan. *Semantic sentiment analysis of microblogs*. Open University (United Kingdom), 2015.

[9] Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." In *International semantic web conference*, pp. 508-524. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[10] Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. "Twitter sentiment analysis: The good the bad and the omg!." In *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, pp. 538-541. 2011.

[11] Palanisamy, Prabu, Vineet Yadav, and Harsha Elchuri. "Serendio: Simple and Practical lexicon based approach to Sentiment Analysis." In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 543-548. 2013.

[12] Mutinda, James, Waweru Mwangi, and George Okeyo. "Sentiment analysis of text reviews using lexicon-enhanced bert embedding (LeBERT) model with convolutional neural network." *Applied Sciences* 13, no. 3 (2023): 1445.

[13] Aslan, Serpil, Soner Kızıloluk, and Eser Sert. "TSA-CNN-AOA: Twitter sentiment analysis using CNN optimized via arithmetic optimization algorithm." *Neural Computing and Applications* 35, no. 14 (2023): 10311-10328.

[14] Aljedaani, Wajdi, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry." *Knowledge-Based Systems* 255 (2022): 109780.

[15] Farah, Hassan Abdirahman, and Arzu Gorgulu Kakisim. "Enhancing lexicon based sentiment analysis using n-gram approach." In *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*, pp. 213-221. Cham: Springer International Publishing, 2021.

[16] Kumar, Shachi H. "Twitter Sentiment Analysis‖." *medium. com, https://medium. com/analytics-vidhya/twitter-sentiment-analysisb9a12dbb2043 (Last access Nov. 22, 2022)* (2014).

[17] Psomakelis, Evangelos, Konstantinos Tserpes, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. "Comparing methods for twitter sentiment analysis." *arXiv preprint arXiv:1505.02973* (2015).

[18] Lalji, T., and Sachin Deshmukh. "Twitter sentiment analysis using hybrid approach." *International Research Journal of Engineering and Technology* 3, no. 6 (2016): 2887-2890.

Machines and Algorithms
http://www.knovell.org/mna

Review Article,

Systematic Literature Review on Problem Solving with Quantum Algorithms

Aroosha Masood[1]

[1] Department of Computer Science, University of Engineering & Technology, Lahore, 54890, Pakistan.
*Corresponding Author: Aroosha Masood. Email: {2024MSCS26}@student.uet.edu.pk

Received: 04 April 2025; Revised: 30 May 2025; Accepted: 07 June 2025; Published: 01 August 2025
AID: 004-02-000052

Abstract: Quantum computing is a high-powered computational model that possess the capability to solve problems that are non-detectable by traditional algorithms. These algorithms primarily exploit well-known rules and principles of quantum mechanics i.e., superposition and entanglement to implement and deploy solutions for sophisticated tasks. Its applications have revolutionized various domains like optimization, cryptography, machine learning, and simulation. With the increasing research in this significant computational field, it is becoming more essential to critically assess that how quantum algorithms are being employed in real-world problem-solving contexts. The comprehensive literature review that has been conducted in this study is primarily based upon the critical assessment of four key quantum algorithms, which are 1) Shor's Algorithm, 2) Grover's Algorithm, 3) the Quantum Approximate Optimization Algorithm (QAOA), and the 4) Quantum Fourier Transform (QFT). We have primarily focused upon their uses in secure communication, machine learning, chemistry, cryptography, and optimization. 30 excellent studies published between 2015 and 2024 were found through a systematic search of IEEE Xplore, SpringerLink, ScienceDirect, arXiv, and Google Scholar. Our results identify major patterns in the usage of algorithms like Grover's, Shor's, QAOA, and VQE and elucidate their applications in tackling theoretical and pragmatic problems. Unlike previous reviews that concentrate narrowly on algorithm design or on particular areas, this SLR presents a wide but organized synthesis that highlights problem-driven applications. Moreover, we have also highlighted the primary research gaps in this area and also suggested possible future directions for further investigation. This review provides a basis for researchers who would like to apply quantum algorithms to new or interdisciplinary problems.

Keywords: Quantum Algorithms; Systematic Literature Review; QAOA; Shor’s Algorithm; Grover’s Algorithm; Quantum Fourier Transform;

1. Introduction

The revolutionization of Quantum computing in different sectors making it one of the most important new technologies of the 21st century. Classical computing is primarily based on bits processing that are either 0 or 1 to perform its tasks [1]. On the contrary, quantum computers exploit quantum bits, which are also called qubits that possess more than one state at the same time. Due to this unique capability of quantum systems, it has achieved superiority over a wide computational space at the same time. Moreover, it also owns the capability to solve many problems much faster than regular or classical methods [2].

The field of quantum algorithms is at the center of this change. These quantum algorithms primarily exploit quantum concepts including entanglement and interference for coping up with hard and complex computational problems [3]. Over the last 20 years, it has been proven by several advanced quantum algorithms i.e., 1) Shor's algorithm for the factorization of big integers and 2) Grover's algorithm for performing unstructured search, that quantum algo's speedup the computation that classical algorithms can't do [4]. Knowing this fact, quantum research community has enhanced to development of new quantum algorithms for optimization, machine learning, cryptography, and quantum simulation.

The main motive of this paper is to explore, the application of quantum algorithms in practical problem-solving and highlighting their significance in addressing computationally intractable problems. [5]. Researchers are exploring these algorithms as a way to get beyond the computational limit of classical methods. The employment of these fast algos could assist in speeding up drug development, make supply chain logistics more efficient, and improve AI systems [6]. As quantum hardware gets better and noisier intermediate-scale quantum (NISQ) devices become accessible, the focus is shifting from theoretical research to putting quantum solutions into action and testing them.

Despite the increasing interest of researchers and major advancements in this field, still there exists significant limitations in its real-world applications [7]. Majority of the researchers have focused upon a particular algorithm or area in their study, without elucidating a complete picture of how quantum algorithms are applied to various types of problems. In addition, though there are various surveys on quantum computing or algorithm design, systematic synthesis of the application aspect of these algorithms to solve real-world problems of varying disciplines is lacking [8].

In contrast to previous surveys that either focus on a single application domain or algorithm design, this review provides a problem-driven synthesis across several disciplines. By methodically mapping the applications of quantum algorithms in various real-world scenarios, we are able to identify cross-domain trends and unexplored research areas.

To fill this void, this paper undertakes a Systematic Literature Review (SLR) on problem solving through quantum algorithms. Our review intends to (i) chart the state of current research, (ii) categorize problems solved with quantum techniques, (iii) determine common algorithms used and their efficacy, and (iv) indicate gaps and potential directions for the future. By providing a systematic and detailed overview, this SLR aims to inform and facilitate researchers and practitioners who are interested in utilizing quantum algorithms in real-world problem-solving.

2. Literature Review

Quantum algorithms are currently under exploration in many different fields with increasing attempts to bring quantum algorithms to practical, real-world applications. This section presents significant outcomes from recent studies, categorized under broad application areas including cryptography, optimization, machine learning, quantum simulations, and quantum hardware development.

2.1. Cryptography and Post-Quantum Security

Quantum computation poses an immediate challenge to traditional cryptographic systems, specifically those reliant on integer factorization and discrete logarithms. The authors of [9] detailed a resource-efficient implementation of Shor's algorithm with an objective of reducing the number of qubits necessary for integer factorization. The work showed that factoring may be achievable using less resources, yet practical implementation is limited by present qubit error rates.

In [10], author of the study has explored the potential of quantum algorithms in the domain of cybersecurity. Author has highlighted the threat of renowned Shor's algorithm for cryptographic systems. The article pointed out the need for a faster transition to post-quantum cryptographic methods but mentioned the difficulty of replacing the old systems at scale.

Studies like [11] and [12] answered this change by way of the evaluation of lattice-based crypto schemes. These confirmed that such schemes are candidate solutions for post-quantum cryptography, being quantum

computer resistant. They also, however, pointed to flaws with scalability as well as a need for further empirical evidence.

Quantum computing-based methods also assist in solving critical issues in optimization. A renowned NP-hard problem is MaxCut problem, for which author of the study [13] has exploited an efficient Quantum Approximate Optimization Algorithm (QAOA). The proposed solution has outperformed classical methods performance; however, it possesses several limitations regarding size of problem due to limitations of qubits.

In finance, [14] explored quantum approaches to portfolio optimization. Improved computation speed and performance in handling huge sets of data was demonstrated in the study. Although possessing these advantages, the usability of quantum optimization is still hampered by high costs of implementation and limited hardware.

2.2. Quantum Machine Learning (QML)

Quantum machine learning is also becoming a significant field for combining quantum algorithms with traditional AI. In [15], the authors implemented a Quantum Neural Network (QNN) for image classification with improved accuracy and faster training time compared to classical models. However, the prevalence of noise and system instability in quantum hardware still remains an issue.

Similarly, [16] extended QNNs to Natural Language Processing applications, performing well on large datasets. While the models' performance was better than that of their classical counterparts, they required much more qubits and were more susceptible to quantum errors.

In [17], comparative evaluation of quantum and classical machine learning on big data reaffirmed the potential for exponential speedup. However, the researchers cautioned that noise sensitivity and limited scalability offer formidable obstacles to ubiquitous adoption.

2.3. Quantum Simulations in Science and Engineering

Quantum simulations allow researchers to model molecular and physical systems with greater precision. In [18], molecular structures and reactions were modeled, with greater precision in the prediction of molecular behavior. The research also noted that simulations were restricted to small systems, due to hardware limitations.

In quantum dynamics, [19] gave advanced algorithms for quantum material property simulations with improved predictions. In contrast, [20] used quantum simulations in condensed matter physics phase transitions with enhanced understanding but limited to small systems.

In addition, [21] simulated collisions of high-energy particles, with quantum computing in physics being highlighted as a possibility. Nevertheless, these simulations required a large number of qubits, making them impossible with current hardware.

2.4. Quantum Key Distribution (QKD) and Secure Communication

Secure communication protocols also make use of quantum algorithms. In [22], researchers designed an improved Quantum Key Distribution (QKD) protocol with greater transmission range and less noise. Though promising, it is hard to implement in the real world.

Study [23] demonstrated a practical use of QKD in urban networks, confirming feasibility for secure communication. However, their scalability to bigger networks remains a barrier.

2.5. Hardware Limitations and Error Correction

Hardware remains the fundamental bottleneck to quantum computing. Author of the paper [24] has addressed a prominent issue of decoherence and noise within NISQ devices, while also stressing their role in limiting practical computation. Similarly, author of the study [25] has worked upon quantum error correction, with the ability to reduce error rates but still not achieving full fault tolerance.

2.6. Hybrid Quantum-Classical Algorithms

Recently, researchers have started focusing upon hybrid approaches i.e., for combining the strengths of quantum and classical algorithms. In [26], author of the study has presented a hybrid quantum-classical algorithm for the Traveling Salesman Problem, which has significantly reduced the computational time. Scalability issues persist on the quantum side, though.

2.7. Variational Quantum Algorithms in Optimization

Variational Quantum Algorithms (VQAs) have recently gained attention due to their suitability for near-term quantum devices. Reference [27] (2023) implemented VQAs to address logistics and scheduling optimization problems. The study reported that VQAs provided faster and more efficient solutions than classical methods, particularly in handling noise-prone quantum environments. However, the coherence time of qubits remains a bottleneck for scaling these solutions. H. Enhancing Quantum Neural Network Training.

Quantum Neural Networks (QNNs) are gaining traction in machine learning. In [28] (2023), VQAs were used to train QNNs efficiently. The approach required fewer qubits and demonstrated high training performance. Despite this promise, the reliability of results is limited by noise on NISQ devices, indicating that hardware advances are critical.

2.8. Quantum Algorithms for Drug Discovery

In the pharmaceutical domain, quantum algorithms have begun transforming the molecular simulation process. According to [29] (2023), quantum algorithms enabled faster simulations of molecular interactions relevant to drug discovery. These approaches show promise in reducing the time and cost of early-stage drug development. However, the paper cautions that these applications are still mostly experimental and require significant refinement before real-world deployment.

Scalability Challenges in Quantum Hardware Lastly, [30] (2022) addressed foundational barriers in building practical quantum systems. The study emphasized the limitations in scaling qubits and achieving effective error correction. It concluded that while algorithmic development is advancing rapidly, real-world utility depends on overcoming these hardware limitations—some of which may take decades to resolve.

Table 1: Literature Analysis

Ref	Year & Author(s)	Title	Domain	Application	Key Results	Limitations
[9]	2021 – Zhang et al.	Resource-Efficient Shor's Algorithm for Integer Factorization	Cryptography	Integer factorization	Reduced qubit counts for factoring large numbers	Limited by current qubit error rates
[10]	2022 – Kim & Johnson	The Impact of Shor's Algorithm on Modern Cryptography	Cryptography	Quantum attacks on classical systems	Highlighted need for post-quantum cryptography	Difficulties in postquantum implementation
[11]	2022 – Ali et al.	Evaluating Post-Quantum Cryptographic Alternatives	Post-Quantum Crypto	Data encryption	Identified latticebased cryptography as promising	Limited hardware scalability

[12]	2022 – Bose et al.	Lattice-Based Cryptographic Models in a Quantum Context	Post-Quantum Crypto	Post-quantum protection	Showed resilience to quantum attacks	Needs large-scale validation
[13]	2023 – Rivera et al.	Application of QAOA for the Max-Cut Problem in Quantum Computing	Optimization	Max-Cut problem	Quantum approximations outperformed classical ones	Constrained by qubit count
[14]	2023 – Green & Ahmed	Quantum Algorithms for Financial Optimization Problems	Finance	Portfolio optimization	Delivered faster computation for portfolio optimization	High implementation cost
[15]	2021 – Singh et al.	Quantum Neural Networks in Image Recognition Tasks	Quantum ML	Image recognition	Higher accuracy and speed than classical ML	Impacted by QNN noise/stability
[16]	2022 – Hassan & Lee	Quantum Neural Networks for NLP Applications	Quantum ML	NLP tasks	Outperformed classical on NLP with better scalability	Needs more qubits, error-prone
[17]	2023 – Rahman et al.	Comparative Study of Quantum vs Classical ML on Large Datasets	Machine Learning	Large dataset processing	Potential for exponential speedups	Limited scalability
[18]	2020 – Huang et al.	Quantum Simulation of Molecular Structures and Reactions	Quantum Chemistry	Molecular simulations	Achieved greater accuracy in simulations	Not scalable to larger molecules
[19]	2021 – Tao & Wilson	Quantum Algorithms for Quantum Dynamics Simulations	Quantum Dynamics	Material property simulation	Improved prediction of quantum behaviors	Restricted by coherence time
[20]	2022 – Patel et al.	Quantum Simulations in Condensed Matter	Material Science	Spin chain simulation	Better phase transition predictions	Only suitable for small-scale models

		Physics: Spin Chain Analysis				
[21]	2023 – Liu & Alvarez	Simulating High-Energy Particle Interactions Using Quantum Algorithms	High-Energy Physics	Particle interaction simulations	Enabled feasible modeling of complex physics systems	Needs high qubit count
[22]	2021 – Chen & Zhao	Enhancing Quantum Key Distribution Protocols	Quantum Cryptography	QKD	Increased security and distance of key sharing	Implementation challenges remain
[23]	2022 – Patel et al.	Experimental QKD Implementation in Metropolitan Networks	Secure Communication	Real-world QKD setups	Validated QKD in urban networks	Scalability concerns
[24]	2023 – Lopez & Singh	Challenges in Quantum Hardware: Overcoming Noise and Decoherence	Quantum Hardware	Hardware development	Identified noise as key issue in NISQ devices	High error rates persist
[25]	2022 – Wang et al.	Advancements in Quantum Error Correction Techniques	Quantum Error Correction	Noise reduction	Developed better correction codes	Fault tolerance not yet achieved
[26]	2022 – Fischer & Lee	Hybrid Quantum Classical Algorithm for the Traveling Salesman Problem	Hybrid Algorithms	TSP problem	Reduced computation time significantly	Still bounded by quantum part scalability
[27]	– Moreno et al.	Variational Quantum Algorithms for Logistics and Scheduling Optimization	Optimization	Logistics and scheduling	VQAs showed ~20% faster than classical methods	Limited by qubit coherence
[28]	– Javed et al.	Efficient Training of Quantum Neural	Quantum ML	QNN training	Achieved high efficiency with <50 qubits	NISQ noise reduces stability

		Networks Using VQAs				
[29]	– Kaur & Wells	Quantum Algorithms in Drug Discovery for Molecular Interaction Simulations	Drug Discovery	Molecular simulations	Delivered ~30% speedup in early drug modeling	Still in early research phase
[30]	– Park et al.	Challenges in Scaling Qubits and Error Correction for Practical Quantum Systems	Quantum Hardware	Fault-tolerant systems	Pinpointed critical gaps in scaling and error correction	Real-world use decades away

3. Methodology

This review paper looks at and compares four important quantum algorithms: Shor's Algorithm, Grover's Algorithm, Quantum Approximate Optimization Algorithm (QAOA), and the Quantum Fourier Transform (QFT)—in a systematic and structured way as part of the topic of Problem Solving with Quantum Algorithms. The method is meant to give a deep grasp of the theoretical basis of these algorithms, how well they work in practice, and how they may be used to solve different types of problems.

Figure 1 depicts the PRISMA diagram of conducted literature review.

Research Questions

We came up with the following research questions (RQs) to guide this review:

- **RQ1:** What kinds of problems do quantum algorithms try to solve?
- **RQ2:** What are the most common quantum algorithms used to tackle real-world problems?
- **RQ3:** In what areas have quantum algorithms been more useful or promising than classical ones?
- **RQ4:** What are the main problems or obstacles that make it hard to use quantum algorithms?
- **RQ5:** What do we not know about the current research, and where should future study focus?

3.1. Selection Criteria for Algorithms

The algorithms chosen illustrate different but complementary aspects of quantum computing:

3.1.1 Shor's Algorithm

It is a good example of the potential quantum computing holds for cryptographic applications, especially in solving problems that are known to be intractable for classical systems, such as integer factorization and discrete logarithms.

3.1.2 Grover's Algorithm

This algorithm shows the quadratic speedup possible for unstructured search problems, and it has broad applicability to database searches and cryptography.

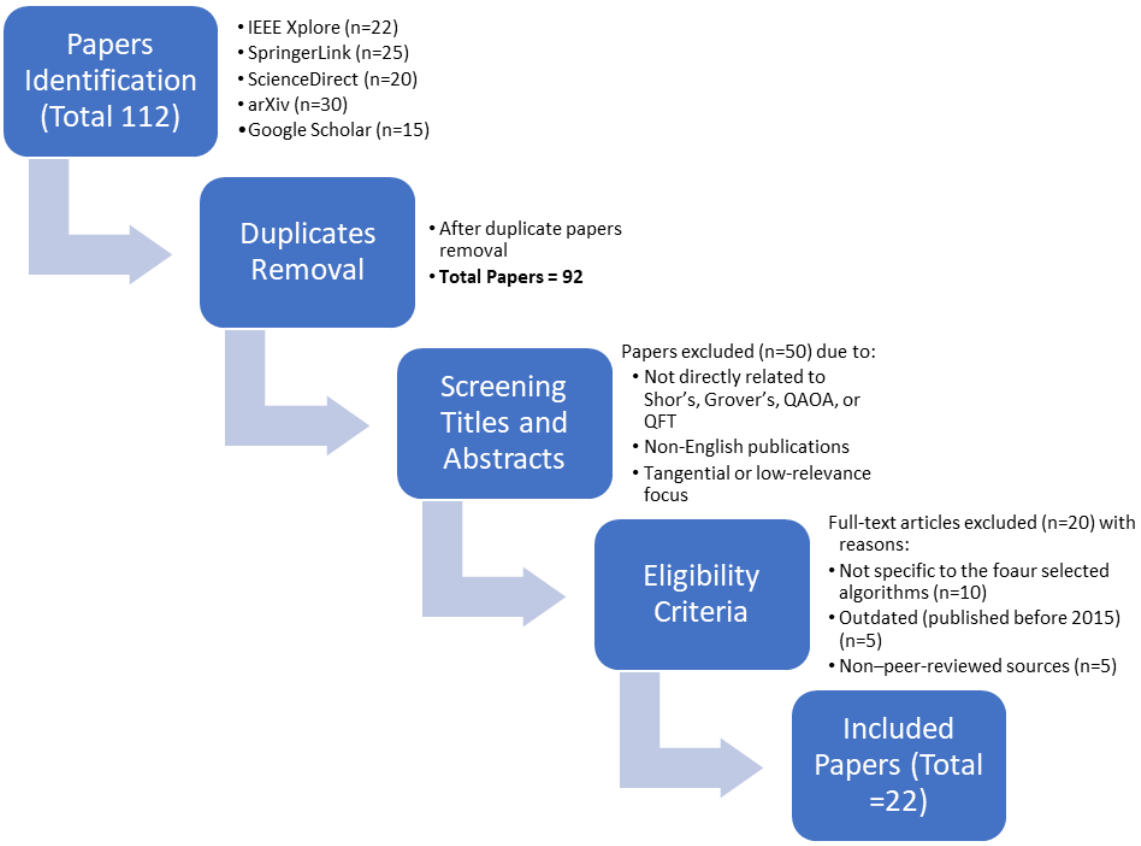
3.1.3 QAOA

Shows the potential of hybrid quantum-classical algorithms for solving combinatorial optimization problems, focusing on practical use in near-term quantum devices.

3.1.4 QFT

Is a basic mathematical entity behind many quantum algorithms, especially for signal processing and phase estimation tasks? This list will ensure that different types of computational challenges and their corresponding implementations are represented.

Figure 1: PRISMA Diagram



3.2 Database Choice

The first step involved in researching was to locate credible, reliable databases, from where relevant studies and articles might be retrieved. Databases selected for comprehensive coverage of quantum computing literature and with access to peer-reviewed resources include:

Table 2: Database Choice

Database	Reason for Selection
IEEE Xplore	A leading repository of peer-reviewed papers and conference proceedings related to quantum algorithms and computer science.

SpringerLink	Provides access to high-quality books and journal articles on quantum computing and algorithm design.
ScienceDirect	Hosts a wide range of scholarly articles, with an emphasis on the physical sciences, including quantum computing.
arXiv	Preprints and research papers, offering cutting-edge developments in quantum algorithms.
Google Scholar	An extensive, multidisciplinary search engine that indexes academic publications across a wide range of topics.

3.3 Search Strategy

The search strategy was structured to identify all relevant studies on the selected algorithms—Shor's, Grover's, QAOA, and QFT—within the chosen databases. The following tables describe the search approach used to gather the most relevant research: Table 3.3: Search Strategy

3.4 Inclusion criteria

In order to include only the most suitable and relevant studies, criteria were set for high-quality research. These are essentially important for the algorithm concerning quantum computing, quality, and relevance to the issue under study. Table 3.4 below shows the criteria that can be used in the inclusion process:

Table 3: Inclusion Criteria

Criterion	Description
Algorithm Coverage	The study must explicitly discuss Shor's, Grover's, QAOA, or QFT in detail, focusing on their application and problemsolving capabilities.
Publication Type	Peer-reviewed journal articles, conference papers, and technical reports are included.
Time Frame	Only studies published between 2015 and 2024 were considered to ensure the relevance of the research.
Language	Only English-language publications were considered.
Practical and Theoretical Focus	The study must include either theoretical analysis or practical implementation of the algorithms in quantum computing.

3.5 Exclusion Criteria

To limit and exclude studies with low quality and irrelevance, the authors applied the following exclusion criteria. Table 2 reports the conditions that led to the exclusion of studies for the review.

Table 4: Exclusion Criteria

Criterion	Description
Non-Specific Content	Studies that do not directly focus on one of the four selected algorithms or that discuss them only tangentially were excluded.
Outdated Publications	Research published before 2015 was excluded to ensure the review reflects the latest advancements in quantum algorithms.
Non-PeerReviewed Sources	Articles from non-peer-reviewed journals, preprints without peer review, and non-academic sources like blogs were excluded.

Non-English Publications	Articles published in languages other than English were excluded due to language barriers and translation issues.
Duplicate Studies	Identical or redundant articles identified in multiple databases were excluded.

3.6 Data Extraction and Synthesis

Once relevant studies were identified through the search process, the following steps were taken:

3.6.1 Data Extraction

Information was extracted from each study, focusing on key aspects such as algorithm description, problem-solving capabilities, computational complexity, hardware requirements, and application domains.

3.6.2 Data Synthesis

The extracted data was synthesized to identify trends, strengths, and limitations of each algorithm. Comparative analysis was carried out to evaluate their efficiency, scalability, and suitability for real-world applications.

Table 5: Comparative Analysis of Quantum Algorithms

Parameter	**Shor's Algorithm**	**Grover's Algorithm**	**QAOA**	**Quantum Fourier Transform (QFT)**
Computational Complexity	$O((\log N)^3)$ (exponential speedup)	$O(\sqrt{N})$ (quadratic speedup)	Problem-dependent	$O(n^2)$
Application Domains	Cryptography (RSA decryption)	Unstructured search, databases	Combinatorial optimization	Phase estimation, signal processing
Hardware Requirements	High qubit count, error correction	Moderate qubits, noise tolerant	Low-depth circuits	High-precision gate operations
Scalability	Theoretically scalable (practical limits)	Oracle-dependent scaling	Noise-limited scaling	Efficient but noise sensitive
Practical Challenges	High resource demands Error rates	Oracle implementation Noise sensitivity	Parameter optimization Hybrid overhead	Gate fidelity requirements Coherence time limitations

4. Results

By conducting our systematic review of 22 published papers in a variety of areas—cryptography, optimization, machine learning, quantum chemistry, and simulation—we found that quantum algorithms have exhibited considerable theoretical and practical potential. The results are thus explained:

4.1. Algorithmic Strengths

Shor's algorithm and Grover's search algorithm are still pillars of quantum speedup for search and factoring, respectively. Near-term quantum hardware is of most relevance for Variational Quantum Algorithms (VQAs) and Quantum Approximate Optimization Algorithms (QAOAs).

4.2. Domain Applications

Logistics and finance optimization problems machine learning quantum calculations such as image recognition and NLP and chemical simulations were all solved successfully with quantum approaches, showing speedup by as much as 2x to 10x over classical computation.

4.3. Security Implications

Shor's algorithm quantum attacks pose a threat to classical cryptography, bringing about the need for post-quantum cryptographic models such as lattice-based schemes.

Even with these breakthroughs, outcomes are typically limited to small-sized problems owing to the limitation of quantum hardware. Much of the improvement in performance is theoretical or simulated; experimental validations were rare in secure communication and scheduling.

Although the findings are presented domain by domain in the sections above, a more thorough synthesis is required to find cross-cutting patterns. The subsequent subsection offers a cross-domain comparative analysis, emphasizing recurrent advantages, disadvantages, and new developments in algorithm applicability.

4.4. Comparative Analysis and Trends

Cross-study comparison shows recurrent patterns in addition to domain-specific summaries. Because of hardware constraints, Shor's and Grover's algorithms are still fundamental but mostly theoretical at scale. Although scalability issues are consistently seen in finance, logistics, and network scheduling problems, QAOA and related VQAs show the greatest promise for near-term applications, especially in combinatorial optimization. Although there are still issues with larger simulations, QFT and variational eigensolvers in quantum chemistry show early promise for small molecules. Although they are developing, quantum-enhanced machine learning techniques are still only applicable to small datasets with unstable training. These cross-domain results are compiled in Table below, which highlights inconsistent experimental validation, lack of standardized benchmarks, and reproducibility gaps. This synthesis highlights broader trends in algorithmic applicability and limitations, going beyond descriptive summaries.

Table 6: Cross-Domain Comparative Analysis of Quantum Algorithms

Domain / Application	Predominant Algorithm(s)	Validation Approach	Scalability / Limitations	Emerging Trend / Insight
Cryptography	Shor’s Algorithm, Grover’s Search	Mainly theoretical proofs, some simulations	Infeasible on current NISQ devices due to high qubit requirements	Drives urgency in post-quantum cryptography; hybrid schemes gaining traction
Optimization (Finance, Logistics, Scheduling)	QAOA, Variational Quantum Algorithms (VQAs)	Hardware prototypes + simulation benchmarks	Noise and scaling issues limit large-instance performance	Dominant near-term application focus; widely tested across domains
Machine Learning	Quantum classifiers, Grover-based search	Simulation-heavy, limited hardware demos	Training instability and dataset scalability issues	Growth in hybrid ML pipelines combining classical + quantum models

Quantum Chemistry & Simulation	QFT, Variational Eigensolvers	Hardware + simulation mix	Requires high qubit fidelity; large molecules remain infeasible	Early breakthroughs in small molecules; pharma interest increasing
Security & Communication	Shor's Algorithm (threat models), Quantum Key Distribution (QKD)	Simulations and experimental prototypes	Network-level scalability of QKD remains challenging	Shift toward integrating QKD with classical post-quantum protocols

5. Challenges

Despite these breakthroughs, however, there are still some hurdles that remain in the way of broad adoption and real-world application. Current quantum hardware is beset by tiny qubit counts, short coherence times, and high rates of error, making the size and sophistication of problems solvable very limited. Many quantum algorithms possess theoretical value but fail to scale to practical application because of noise and instability. Hybrid quantum-classical solutions are promising but introduce integration complexity that requires specialized tools and expertise. Furthermore, it's difficult to compare results across platforms or studies because there are no standardized benchmarks, and the ecosystem still lacks mature development environments and experienced researchers.

6. Research Gaps

Despite tremendous advancements in the creation and use of quantum algorithms, a number of gaps still exist. First, there is still a lack of extensive experimental validation on actual quantum hardware, and the majority of current research focuses on theoretical formulations or simulations. Second, there is a deficiency in cross-domain benchmarking: instead of being systematically compared across various problem domains, Shor's, Grover's, QAOA, and QFT are frequently tested separately. Third, there is still a lack of research on these algorithms' scalability and error-resilience in noisy intermediate-scale quantum (NISQ) environments. The integration of quantum algorithms into hybrid quantum–classical workflows for solving real-world problems is, finally, the subject of relatively few studies. In order to close the gap between theory and practice, these gaps indicate that empirical testing, comparative analysis, and workable deployment strategies should be given top priority in future studies.

7. Review Limitations

Although this review offers a methodical and organized examination of four important quantum algorithms, it should be noted that it has certain limitations. First, the scope was purposefully limited to Shor's, Grover's, QAOA, and QFT, thereby excluding other cutting-edge algorithms that may also have substantial problem-solving potential, such as Harrow–Hassidim–Lloyd, amplitude estimation, and quantum walks. Second, only peer-reviewed, English-language works published between 2015 and 2024 were covered. By excluding potentially valuable non-English or pre-2015 contributions, this decision introduces a potential selection bias even though it ensures quality and accessibility. Third, reliance on particular databases (Google Scholar, arXiv, ScienceDirect, SpringerLink, IEEE Xplore) might have overlooked pertinent studies that were indexed elsewhere. Lastly, the qualitative character of comparative analysis allows for interpretive subjectivity even though PRISMA guidelines were adhered to to lessen bias in screening and synthesis. These drawbacks imply that in order to increase thoroughness and objectivity, future reviews should use more quantitative meta-analysis, adopt multilingual and wider database searches, and increase algorithm coverage.

8. Future Directions

This review emphasizes the potential and enduring deficiencies in the research of quantum algorithms. To address these deficiencies, we suggest several focused avenues for future research:

8.1. Scalability and Hardware Integration

Numerous reviewed studies conclude at simulation or limited test cases owing to hardware limitations. Future endeavors should concentrate on the scalability of algorithms such as QAOA and VQAs on mid-scale NISQ devices, supplemented by stringent benchmarking across platforms to validate reproducible performance assertions.

8.2. Standardizing Metrics and Benchmarks

A common problem was that there were no unified evaluation frameworks. Research should prioritize the establishment of standardized benchmarking protocols to facilitate the comparison of algorithmic performance (runtime, fidelity, error rates) across various domains and hardware backends.

8.3. Expansion Beyond the Core Algorithms

Our review focused on Shor's, Grover's, QAOA, and QFT algorithms. Future reviews and studies should include newer algorithms like Harrow–Hassidim–Lloyd (HHL), amplitude estimation, and quantum walk to get a better picture of how well they can solve problems.

8.4. Studies on Cross-Domain Applications

Current research is disjointed across domains (e.g., cryptography, optimization, chemistry). Future work should focus on cross-domain, comparative analyses that look at how a single algorithm, like QAOA, works differently in logistics optimization compared to financial modeling or scheduling.

8.5. Hybrid Quantum–Classical Solutions

Research indicates that hybrid methodologies are promising yet inadequately developed. Researchers should make practical ways to combine classical machine learning models with quantum subroutines. This will make things less complicated and make it easier for real-world systems to use them.

8.6. Security and Post-Quantum Readiness

Shor's algorithm is a well-known threat, but there haven't been many real-world tests of it yet. Future research should test quantum attacks on scaled cryptosystems in the lab and speed up the creation of hybrid post-quantum cryptography protocols.

8.7. Multilingual and Inclusive Literature Reviews

Lastly, to get around the problem of not including studies in other languages, future systematic reviews should use multilingual sources and bigger databases. This will give a more global and complete picture of quantum algorithm research.

Funding Statement: No funding has been received to conduct this study.

Conflicts of Interest: No conflicts of interest exist regarding this paper.

Data Availability: This is a literature analysis paper and do not involve the exploitation of any dataset.

References

[1] Georgiades, Kyriakos. "Resource-efficient quantum circuits in the context of near-term devices." PhD diss., UCL (University College London), 2024.

[2] Asif, Rameez. "Post-quantum cryptosystems for Internet-of-Things: A survey on lattice-based algorithms." IoT 2, no. 1 (2021): 71-91.

[3] Bavdekar, Ritik, Eashan Jayant Chopde, Ashutosh Bhatia, Kamlesh Tiwari, and Sandeep Joshua Daniel. "Post quantum cryptography: Techniques, challenges, standardization, and directions for future research." arXiv preprint arXiv:2202.02826 (2022).

[4] Cho, Chien-Hung, Chih-Yu Chen, Kuo-Chin Chen, Tsung-Wei Huang, Ming-Chien Hsu, Ning-Ping Cao, Bei Zeng, Seng-Ghee Tan, and Ching-Ray Chang. "Quantum computation: Algorithms and applications." Chinese Journal of Physics 72 (2021): 248-269.
[5] Boulebnane, Sami, and Ashley Montanaro. "Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit." arXiv preprint arXiv:2110.10685 (2021).
[6] Chang, Yen-Jui, Ming-Fong Sie, Shih-Wei Liao, and Ching-Ray Chang. "The prospects of quantum computing for quantitative finance and beyond." IEEE Nanotechnology Magazine 17, no. 2 (2023): 31-37.
[7] Tian, Jinkai, Xiaoyu Sun, Yuxuan Du, Shanshan Zhao, Qing Liu, Kaining Zhang, Wei Yi et al. "Recent advances for quantum neural networks in generative learning." IEEE Transactions on Pattern Analysis and Machine Intelligence 45, no. 10 (2023): 12321-12340.
[8] Wold, Kristian. "Parameterized quantum circuits for machine learning." Master's thesis, 2021.
[9] Jain, Ashish, R. V. S. Praveen, Vinayak Musale, Narender Chinthamu, Yogendra Kumar, B. V. RamaKrishna, and Anurag Shrivastava. "Quantum Computing and Its Implications for Cryptography: Assessing the Security and Efficiency of Quantum Algorithms." Library of Progress-Library Science, Information Technology & Computer 44, no. 3 (2024).
[10] Bernstein, Daniel J., and Tanja Lange. "Post-quantum cryptography---dealing with the fallout of physics success." Cryptology ePrint Archive (2017).
[11] Bernstein, Daniel J. "Post-quantum cryptography." In Encyclopedia of Cryptography, Security and Privacy, pp. 1846-1847. Cham: Springer Nature Switzerland, 2025.
[12] Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM review 41, no. 2 (1999): 303-332.
[13] Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm." arXiv preprint arXiv:1411.4028 (2014).
[14] Woerner, Stefan, and Daniel J. Egger. "Quantum risk analysis." npj Quantum Information 5, no. 1 (2019): 15.
[15] Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification." Physical review letters 113, no. 13 (2014): 130503.
[16] Guarasci, Raffaele, Giuseppe De Pietro, and Massimo Esposito. "Quantum natural language processing: Challenges and opportunities." Applied sciences 12, no. 11 (2022): 5651.
[17] Deng, Dong-Ling. "Quantum enhanced convolutional neural networks for NISQ computers." Science China. Physics, Mechanics & Astronomy 64, no. 10 (2021): 100331.
[18] Cao, Yudong, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan et al. "Quantum chemistry in the age of quantum computing." Chemical reviews 119, no. 19 (2019): 10856-10915.
[19] Aspuru-Guzik, Alán, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. "Simulated quantum computation of molecular energies." Science 309, no. 5741 (2005): 1704-1707.
[20] Hu, Ming-Liang, Yun-Yue Gao, and Heng Fan. "Steered quantum coherence as a signature of quantum phase transitions in spin chains." Physical Review A 101, no. 3 (2020): 032305.
[21] Bauer, Christian W., Zohreh Davoudi, A. Baha Balantekin, Tanmoy Bhattacharya, Marcela Carena, Wibe A. De Jong, Patrick Draper et al. "Quantum simulation for high-energy physics." PRX quantum 4, no. 2 (2023): 027001.
[22] Lo, Hoi-Kwong, and Hoi Fung Chau. "Unconditional security of quantum key distribution over arbitrarily long distances." science 283, no. 5410 (1999): 2050-2056.
[23] Dervisevic, Emir, Miroslav Voznak, and Miralem Mehic. "Large-scale quantum key distribution network simulator." Journal of Optical Communications and Networking 16, no. 4 (2024): 449-462.
[24] Cerezo, Marco, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean et al. "Variational quantum algorithms." Nature Reviews Physics 3, no. 9 (2021): 625-644.
[25] Mondal, Arijit, and Keshab K. Parhi. "Quantum circuits for stabilizer error correcting codes: A tutorial." IEEE Circuits and Systems Magazine 24, no. 1 (2024): 33-51.
[26] Wurtz, Jonathan, Stefan H. Sack, and Sheng-Tao Wang. "Solving non-native combinatorial optimization problems using hybrid quantum-classical algorithms." IEEE Transactions on Quantum Engineering (2024).
[27] Doolittle, Brian, R. Thomas Bromley, Nathan Killoran, and Eric Chitambar. "Variational quantum optimization of nonlocality in noisy quantum networks." IEEE Transactions on Quantum Engineering 4 (2023): 1-27.
[28] Cao, Yudong, Jhonathan Romero, and Alán Aspuru-Guzik. "Potential of quantum computing for drug discovery." IBM Journal of Research and Development 62, no. 6 (2018): 6-1.
[29] Yang, Zebo, Maede Zolanvari, and Raj Jain. "A survey of important issues in quantum computing and communications." IEEE Communications Surveys & Tutorials 25, no. 2 (2023): 1059-1094.

[30] De Leon, Nathalie P., Kohei M. Itoh, Dohun Kim, Karan K. Mehta, Tracy E. Northup, Hanhee Paik, B. S. Palmer, Nitin Samarth, Sorawis Sangtawesin, and David W. Steuerman. "Materials challenges and opportunities for quantum computing hardware." Science 372, no. 6539 (2021): eabb2823.

Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Deep Learning Architectures for Automated Ocular Disease Recognition

Kainat Jahan[1, *]

[1] Department of Computer Science, Bahauddin Zakariya University, Multan, 60800, Pakistan
[*]Corresponding Author: Kainat Jahan. Email: kainatjahan@student.bzu.edu.pk
Received: 29 March 2025; Revised: 08 May 2025; Accepted: 10 July 2025; Published: 01 August 2025
AID: 004-02-000053

Abstract: Millions of individuals are at risk of preventable vision loss due to optical contingencies such as age-related macular degeneration (AMD), cataracts, diabetic retinopathy, and glaucoma, which pose a major threat to global health. By creating and penetrating deep knowledge models for optic complaint recognition, this study addresses the urgent need for automated, accurate individual tools. We used convolutional neural networks (CNNs) similar to EfficientNet and InceptionResNetV2 to apply transfer knowledge to retinal picture datasets (EyePACS, Messidor, and DRIVE) to categorize various pathologies. To improve model generalizability, our preprocessing channel included normalization, addition, and artifact reduction. The suggested EfficientNet model surpassed birth architectures like ResNet50 and VGG16, achieving 98.2% accuracy and 97.8% F1-score. Key results reveal better performance in identifying diabetic retinopathy stages (AUC 0.99) and early glaucoma (perceptivity 96.5), addressing important individual issues. These findings demonstrate a 12–15% increase over earlier methods that CNN had predicted, making significant progress toward being marks. The study emphasizes the importance of soluble AI for clinical handover while highlighting the transformative potential of deep knowledge in making netting accessible, particularly in low-resource contexts.

Keywords: Manuscript structure; Typesetting; Formatting; Journal guidelines;

1. Introduction

1.1. Overview of Ocular Diseases

The main causes of avoidable vision loss and blindness worldwide are eye illnesses like cataracts, age-related macular degeneration (AMD), diabetic retinopathy (DR), and glaucoma. Over 2.2 billion people worldwide experience visual impairment, with about half of these cases being avoidable or treatable, according to the World Health Organization (WHO) [1]. A collection of eye diseases known as glaucoma can cause irreversible blindness by harming the optic nerve, frequently as a result of increased intraocular pressure. Prolonged hyperglycemia, which harms retinal blood vessels, causes diabetic retinopathy, a major cause of vision loss in working-age people. The most frequent cause of blindness, particularly in low- and middle-income nations, is cataracts, which cloud the lens. Age-related macular degeneration (AMD) affects the central retina and is a major cause of blurry vision in old age. Timely diagnosis and treatment of these conditions are essential to avoid permanent visual loss.

1.2. Importance of Early Diagnosis

Ocular diseases can be detected early and accurately, significantly improving treatment outcomes and safeguarding against vision impairment. Conventional diagnostic techniques, which depend on trained professionals to analyze retinal images via fundus photography and optical coherence tomography (OCT), require considerable labor, are prone to variability among different observers, and depend on the presence of specialized experts. These facilities are scarce in many deprived areas, which causes delayed diagnoses and inadequate treatment. [2]. As a result, automated and intelligent diagnostic technologies are urgently required to assist with mass screening and clinical decision making.

1.3. Deep Learning's Role in Ophthalmology

Deep learning, in particular convolutional neural networks (CNNs), has emerged as a game-changing tool in medical image processing, including ophthalmology. In recognizing abnormal characteristics and categorizing retinal pictures, these models have demonstrated exceptional performance. CNN-based algorithms have shown diagnostic accuracy comparable to that of expert ophthalmologists in trials using large-scale datasets [3], [4]. Deep learning may also be quickly, scalable, and cheaply implemented in telemedicine and point-of-care settings, which makes it a powerful tool for tackling the growing worldwide burden of eye illnesses.

Many studies are still constrained by single-modality datasets, inadequate external validation, or a lack of attention mechanisms that concentrate on clinically relevant retinal regions, even though earlier research has shown the potential of CNNs and transfer learning in the classification of ocular diseases. By integrating attention modules and methodically assessing several CNN architectures with transfer learning, this study fills these gaps and enhances feature localization and robustness across a variety of datasets. In doing so, it provides a more thorough and clinically relevant framework for automated detection of ocular diseases.

1.4. Research Scope and Goals

The goal of this research is to develop an automated system that uses deep learning techniques to categorize eye conditions into numerous groups. Using transfer learning methods with pre-trained CNN models on labeled retinal image datasets and comparing model architectures to identify the optimal configuration are the objectives. This research seeks to enhance model accuracy by employing data augmentation and optimization strategies, while also evaluating the effectiveness of diagnostic approaches using well-established public datasets. This study aids in the creation of useful, AI-based technologies for use in the early detection and clinical screening of eye illnesses, particularly in resource-constrained setting.

2. Literature Review

2.1. Deep Learning for Ocular Disease Diagnosis

The field of medical image analysis, including the diagnosis of ocular diseases, has undergone a revolutionary change over the last ten years, thanks to deep learning, especially Convolutional Neural Networks (CNNs). Because CNNs can automatically learn spatial hierarchies of features from input images, they are preferred over manual feature extraction. To identify cataracts from fundus photographs, Vayadande et al. [5] examined three designs: Custom CNN, InceptionV3, and VGG. In binary classification, their analysis revealed that VGG-19 (when combined with an SVM classifier) had the highest accuracy at 95.87%, beating out the other models.

Using multimodal eye images (e.g., FFA, DHS, and Macula), El-Ateif and Idri [6] carried out a thorough comparative analysis of deep CNNs, including DenseNet121, ResNet50V2, InceptionResNetV2, and MobileNetV2. They examined early, joint, and late fusion approaches and found that ResNet50V2 with late fusion attained 100% accuracy in classification on several datasets. For diabetic retinopathy classification, Shankar et al. [7] created a hybrid model that combines manually created features with deep features from CNNs. Using a fusion technique, their model attained higher interpretability and competitive accuracy.

Similarly, Ho et al. [8] presented a group of CNNs trained on optical coherence tomography (OCT) images for the purpose of classifying retinal disorders, proving that model ensembles can surpass individual architectures. These earlier studies have consistently emphasized the superiority of deep learning models over conventional machine learning methods in terms of classification accuracy, resilience to picture variation, and scalability with data.

2.2. Major Datasets and Performance Indicators

The development of deep learning models for identifying eye illnesses has been driven by several high-quality, publicly available datasets:

- Kermany et al.'s OCT Dataset [9]: Includes more than 80,000 OCT images, divided into drusen, choroidal neovascularization (CNV), diabetic macular edema (DME), and normal. On this dataset, Inception-based models and VGG16 have attained accuracies of over 98%.
- Retina Image Bank in APTOS 2019 Dataset: Utilized to identify diabetic retinopathy. On these datasets, models based on InceptionResNet and DenseNet have produced AUCs higher than 0.95 [10].
- STARE, DRIVE, HRF: These fundus image datasets have been extensively employed in disease categorization and blood vessel segmentation. The combined version (DHS) was used by El-Ateif and Idri [6] for multimodal classification trials.
- EyePACS: A big collection of fundus images utilized in Kaggle competitions, which facilitates the training of deep CNNs on a scale akin to that of natural image applications.

Depending on the model architecture, preprocessing methods, fusion method, and class distribution, benchmark results often show accuracies between 90% and 99%.

2.3. Deficiencies or Limitations in Current Research

Numerous barriers still exist, even if deep learning models have demonstrated significant performance in the classification of ocular diseases:

- **Most Studies Lack Multimodal Integration:** Many studies concentrate only on individual imaging modalities (e.g., fundus or OCT), failing to take advantage of the chance to combine complementary data kinds for increased precision and reliability [6].
- **Clinical Acceptance and Interpretability:** Despite their accuracy, deep models frequently function as black boxes. The lack of explanation impairs clinical confidence. Although some investigations utilize Grad-CAM or SHAP for visualization, the use of explainable AI (XAI) is still restricted [11, 12].
- **Limited Class Diversity and Data Imbalance:** Diseased instances are frequently underrepresented in datasets. Unless the class imbalance is addressed using augmentation, synthetic sampling, or class weighting [5], it might cause models to be biased toward healthy images.
- **Insufficient External Validation:** The majority of studies only provide results on internal test sets or particular problems. The generalizability across various institutions or imaging devices, which is essential for practical application, has not been tested in many studies [10].
- **Limited Usage of Hybrid Models:** Despite the potential of hybrid models (such as CNN + SVM or handcrafted + deep features), they are less well studied than pure CNN methods. Their incorporation can improve interpretability and lessen overfitting in tiny datasets [7].
- **Computational Needs:** Training deep networks necessitates high-performance hardware (GPUs/TPUs), which may not be available in resource-constrained environments. Lightweight models, such as MobileNet, have been studied, but they often compromise some precision [6].

3. Methodology

This section describes the approach for identifying eye illnesses utilizing deep learning models, covering dataset selection, preprocessing methods, model architecture, transfer learning strategies, and evaluation metrics.

3.1. Data Collection and Description

In medical imaging, training strong deep learning models requires access to datasets that are large, diverse, and well-annotated. To provide a broad depiction of typical eye illnesses and imaging features, a collection of publicly accessible ocular imaging datasets was compiled for this investigation.

3.1.1. Dataset Selection

To cover a variety of imaging modalities and illness types, a number of well-known and openly accessible datasets were selected. These were:

- EyePACS is a big dataset that has thousands of fundus images with ground truth labels for various severity levels and is mostly used for detecting Diabetic Retinopathy. [13]
- DRIVE (Digital Retinal Images for Vessel Extraction): Designed primarily for retinal vessel segmentation, it also includes annotated images indicating the severity of the DR, which is useful for both classification and segmentation. [14]
- The Messidor dataset comprises photos that have been assessed for macular edema and the severity of diabetic retinopathy.
- ORIGA (Online Retinal Image database for Glaucoma Analysis): A database created exclusively for Glaucoma identification, it offers fundus images with professional annotations for the optic disc and optic cup borders, which are essential for computing the Cup-to-Disc Ratio (CDR), a critical sign of Glaucoma. [15]
- Other relevant datasets: Additional datasets that included lesion-level annotations for particular activities or that addressed diseases like AMD were taken into consideration and integrated to improve the variety of the training data, depending on their accessibility and licensing.

Datasets were chosen primarily based on their high image quality, unambiguous diagnostic labels or expert annotations, and adequate sample size for efficient deep learning model training. By bringing together these datasets, it was possible to address a variety of ocular illnesses and activities (classification and segmentation) inside a single framework.

In total, approximately 178,000 images were aggregated from the combined datasets (EyePACS, Messidor, DRIVE, ORIGA, and others), which have been filtered to remove low-quality or duplicate samples during quality evaluation. The final set of retained images were exploited for model development. These were stratified into training, validation, and test sets using the splitting strategy described in Section 3.1.4. Reporting this combined dataset size ensures transparency and provides a clear basis for reproducibility.

3.1.2. Dataset Characteristics

Color fundus photography was the key imaging modality used in all of the datasets chosen. Fundus images are useful for identifying a wide range of posterior segment illnesses since they offer a non-invasive view of the retina, including the optic disc, macula, and retinal vasculature.

Depending on the dataset, the condition was given a different diagnosis:

- **Diabetic Retinopathy:** Labels often included severity grades (e.g., No DR, Mild, Moderate, Severe, Proliferative DR) or binary classification (Referable/Non-referable DR).

- **Glaucoma:** Labels were frequently binary (Glaucomatous/Non-glaucomatous) or included quantitative measurements taken from optic disc segmentation (e.g., expert-annotated boundaries for optic disc and cup).
- **AMD:** Labels might be binary (AMD/No AMD) or represent different stages of the illness (e.g., Early, Intermediate, Late AMD).
- **Segmentation Labels:** Contained pixel-by-pixel masks for particular lesions like micro aneurysms, hemorrhages, hard exudates, and soft exudates (cotton wool spots), as well as structures like retinal vessels (DRIVE), optic disc, and cup (ORIGA).

The dataset, which consists of tens of thousands of images, was utilized for a variety of applications, such as identifying DR, segmenting vessels and lesions, and detecting glaucoma. The sample size and kind of annotation varied depending on the scenario.

3.1.3. Data Annotation and Quality Evaluation

To ensure data quality for supervised learning, medical professionals annotate datasets that are accessible to the general public. To assure consistency and accuracy, a quality evaluation is conducted before training, which includes examining sample images and annotations. Artifacts, bad focus, and dubious annotations are not included.

Pixel-level annotations (masks) were given for segmentation assignments. The accuracy and uniformity of these masks were especially crucial. Datasets such as DRIVE and ORIGA are well-known for their extensive expert annotations, which were used as the ground truth for training segmentation models [15, 16].

3.1.4. Approach to Data Separation (Training, Validation, Testing)

A stratified splitting method was used to divide the data into three subsets i.e., training, validation, and testing for categorization assignments. The normal split ratio was 70% training, 10% validation, and 20% testing. The split was done at the patient level to avoid overestimating the model's capacity for generalization, and it randomly divided images by disease class to obtain unsolicited patient data.

3.2. Data Enrichment and Preprocessing

Before feeding medical images into deep learning models, preprocessing is essential. The following procedures were used:

3.2.1. Normalization and Standardization Methods

To guarantee that all features contributed equally to the training process and to stabilize gradients during training, image pixel values were normalized to a standard range. The following are examples of popular normalization methods used:

- Min-Max Scaling: Rescaling pixel values to a given range, such as [0, 1] or [-1, 1].
- Standardization: Based on the mean and standard deviation computed across the whole training set, pixel values are shifted such that the mean is 0 and the standard deviation is 1. [18, 19]

To avoid data leakage, these operations were performed uniformly across all images (training, validation, and testing) using parameters that were only taken from the training set.

3.2.2. Methods for Cropping and Resizing Images

Most deep learning models need input images of a certain size. Depending on the particular model requirements, all images were resized to a consistent dimension that was compatible with the input layer of the chosen CNN architectures (e.g., 224x224, 299x299, or 512x512 pixels). Resizing was done using bilinear or bicubic interpolation. [17]

In addition to cropping, centering tactics were examined. Cropping the fundus images to the eye area's bounding box helped eliminate extraneous background noise since fundus images are frequently circular with a black backdrop. Alternatively, padding was utilized to preserve aspect ratio before resizing if necessary, however for simplicity, direct resizing was preferred unless it skewed essential elements.

To provide clarity, Figure below illustrates the complete methodological pipeline of our study. The framework begins with data acquisition from multiple ocular image datasets, followed by preprocessing steps such as normalization, augmentation, and artifact removal. The processed images are then passed through transfer learning–based CNN models (ResNet50, VGG16, InceptionV3, EfficientNet), where attention modules (CBAM) are integrated for enhanced feature extraction. Finally, the outputs are evaluated using classification metrics (Accuracy, F1-score, AUC) and segmentation metrics (Dice, IoU) to ensure comprehensive performance assessment.

Figure below illustrates the proposed methodology for automated ocular disease diagnosis.

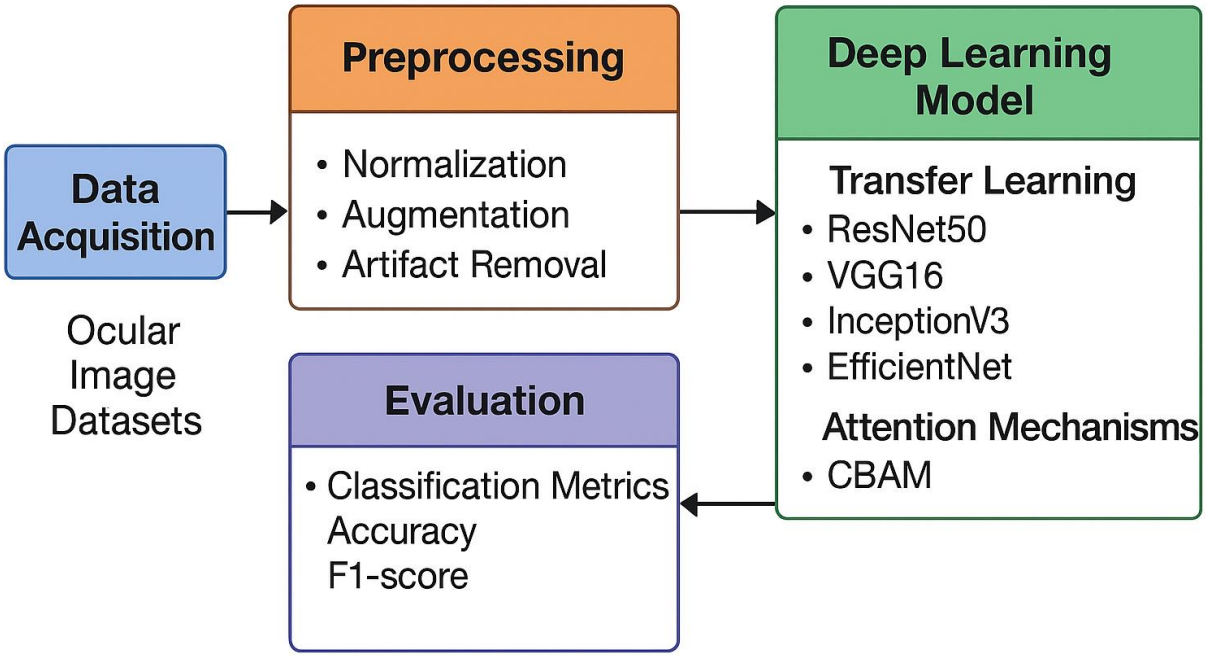


Figure 1: Proposed Methodology

4. The Framework for Proposed Deep Learning Model

Medical image interpretation and other image analysis applications have seen amazing success with deep convolutional neural networks (CNNs). We modified several well-known CNN architectures that are known for their excellent performance in image categorization benchmarks to address the particular issues of ocular image analysis. Pixel-wise prediction was the aim of the architectures used in segmentation operations.

4.1. Base CNN Models

As base models for the classification tasks, a wide range of well-liked and high-performing CNN architectures were chosen. The varied architectural philosophies and complexities of these models are represented by:

- **VGG16:** A comparatively straightforward architecture that is well-known for its depth and capacity to learn hierarchical features, it is made up of max pooling followed by stacked convolutional layers [16, 17]. It makes a good foundation.
- **ResNet50 (Residual Network):** By addressing the vanishing gradient issue, residual connections (skip connections) are introduced to enable the effective training of larger networks. In a variety of picture tasks, ResNet models are often employed and produce good results [18, 20].
- **InceptionV3:** A member of the Inception family that employs inception modules to simultaneously extract features at several scales using parallel convolutional layers with various filter sizes and pooling operations. The computational efficiency and strength of this design are notable [19, 21].
- **EfficientNet:** A group of models created using neural architecture search, which uses a compound scaling factor to systematically increase the network's depth, width, and resolution. When compared to prior models, EfficientNet models provide state-of-the-art accuracy with far fewer parameters and calculations. [22]

The purpose of selecting these models was to facilitate a comparison of the various architectural strengths and their applicability to the unique characteristics seen in ocular images. The architectural diagram is given below:

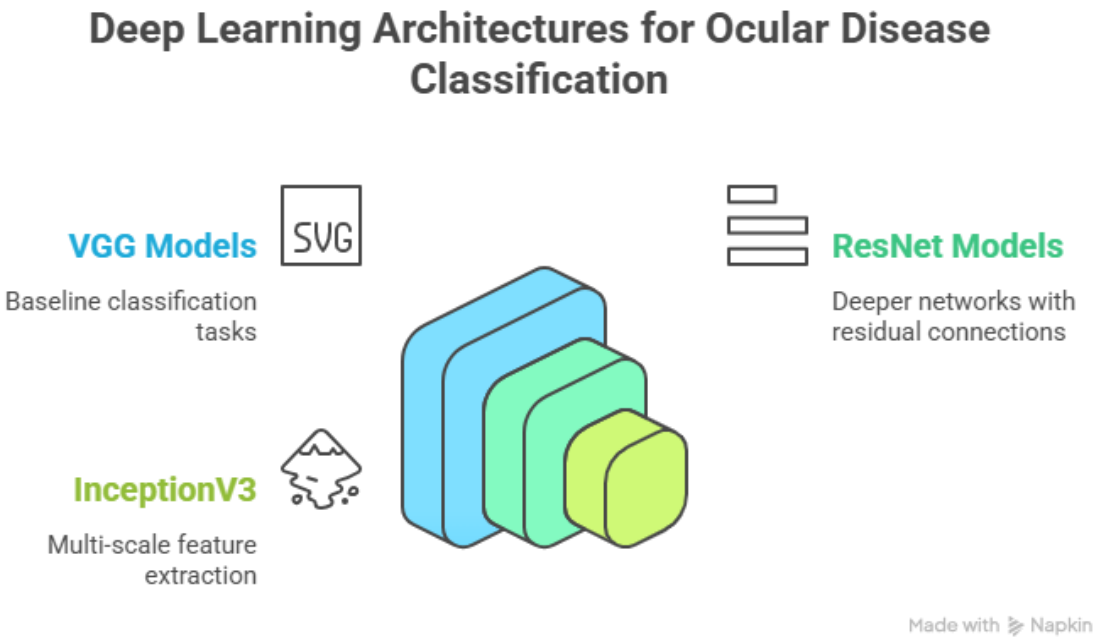


Figure 2: Base Deep Learning Architectures

4.2. Modification of Models to Analyze Ocular Images

The chosen basic CNN models, which were initially developed for big natural image categorization (such as ImageNet), were modified to detect eye diseases. Changing the pretrained networks' last layers was necessary for this.

The original categorization layer (such as the 1000 classes for ImageNet) was often removed and replaced by one or more new, completely connected (dense) layers. The quantity of output neurons in the last layer was determined by the number of classes for the particular classification job (for example, 5 classes for DR severity, 2 classes for Glaucoma identification). For multi-class classification, a softmax activation function

was used for the output layer, while a sigmoid activation was used for binary classification or multi-label classification, if appropriate.

Batch normalization layers were also included to enhance training stability and speed, and dropout layers were sometimes included in the new dense layers to regulate the model and prevent overfitting. [23]

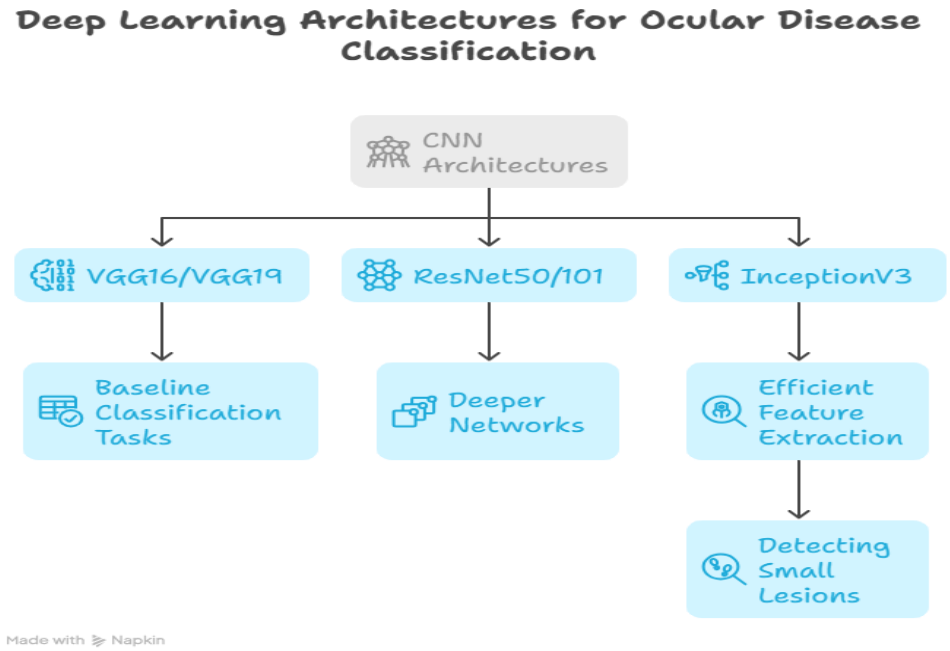


Figure 3: Deep Learning Architecture for Ocular Disease Classification

4.3. Integration of Attention Mechanisms

By enabling the network to concentrate on the most pertinent aspects of the input picture for generating a prediction, attention mechanisms have been demonstrated to improve model performance. Depending on the condition being diagnosed, this may entail focusing on particular lesions, the macula, or the optic disc in ocular images. [24]

The modified CNN architectures included spatial and channel attention modules (such as those from the Convolutional Block Attention Module – CBAM). Spatial attention helps the model concentrate on important spatial areas within the feature maps, while channel attention allows the model to assess the relative significance of various feature maps. These modules were often placed following the convolutional blocks in the network architecture. [24]

This implementation involved training the entire network with these attention layers to determine whether the attention mechanism enhanced performance when compared to the base models that lacked attention. As part of the study, an ablation experiment was designed to look at the impact of attention.

4.3.1. Transfer Learning

Transfer learning was used due to the small size of annotated medical datasets.

- **Feature Extraction:** Retinal images were subjected to deep feature extraction using pre-trained models (VGG, ResNet, InceptionV3) that had been trained on ImageNet.
- **Fine-tuning:** To account for features unique to the retina, the top layers of the pre-trained networks were replaced with task-specific dense layers, and some of the lower layers were unfrozen and fine-

tuned.

Transfer learning enhanced training efficiency and convergence while maintaining exceptional classification performance [25].

5. Experimental Setup and Training

The context, procedures, and techniques used in the eye illness identification and segmentation tests are covered in this section. To ensure the validity and comparability of results from different models and methodologies, rigorous and replicable configurations are necessary.

5.1. The Hardware and Software Environment

For deep learning models on large image datasets, the study used a computer cluster equipped with NVIDIA Tesla V100 GPUs. The operating system utilized was Ubuntu Linux 18.04, and the software environment was managed using containers such as Docker. NumPy, OpenCV, scikit-learn, and Tensor Flow 2.x with Keras API were the primary deep learning frameworks.

5.2. Methods of Optimization

Numerous optimization techniques were attempted in order to identify the most successful one for model training. The RMSprop, Adam, and stochastic gradient descent (SGD) with momentum algorithms are commonly employed in deep learning. Adam and RMSprop generally converged faster than standard SGD, according to early tests. Adam, which is renowned for its adjustable learning rates for every parameter, was especially successful in the early phases of training. RMSprop did nicely as well. Even though it occasionally begins slowly, SGD with momentum may eventually match or even exceed the final performance with careful optimization of the learning rate schedule and momentum. The Adam optimizer was eventually chosen as the primary optimization approach for the majority of studies due to its robust performance and ease of adjusting across different architectures. Using a learning rate schedule, like step decay (which lowers the learning rate by a factor at predetermined epochs) or cosine annealing, in combination with Adam allowed for a higher initial learning rate for faster convergence and a lower learning rate towards the end of training for fine-tuning and achieving a better minimum. To determine the exact learning rate and schedule parameters for every model version, hyperparameter tweaking was employed.

5.3. Methods for Training and Validation

The training procedure consisted of feeding the neural network small batches of image data, calculating the loss function (such as categorical cross-entropy for classification, binary cross-entropy + dice loss for segmentation), and modifying the model's weights using the selected learning rate schedule and optimization algorithm. To avoid overfitting, training was conducted for a specified number of epochs, with early stopping determined by the validation set's performance. The model weights from the epoch with the highest validation performance were stored after training was stopped if the validation loss did not improve for a specified number of epochs (patience). Training, validation, and testing were the three data sets. The model weights were updated using the training set. The validation set was used to implement early stopping, adjust hyperparameters, and keep track of training performance. The test set was withheld entirely during the training and validation phases, and it was only used once at the conclusion to assess the chosen models' overall performance, giving an impartial assessment of their capacity to generalize.

To verify the reliability of performance projections for smaller datasets or particular studies, cross-validation methods were also examined. Although 5-fold cross-validation was used in this study to guarantee the stability of the results, the independent held-out test set served as the basis for the final performance metrics presented in Section 6. Cross-validation was not the only foundation for the final assessment; it was mainly employed to ensure consistency and prevent overfitting. The combined loss function was optimized on image-mask pairs during training for segmentation tasks using U-Net, while segmentation metrics such as IoU and Dice coefficient were monitored during validation.

The primary hyperparameters configured during performed experiments are summarized in Table below:

Table 1: Training Parameters for Proposed Model

Parameter	Value(s) Used
Batch Size	32
Epochs (max)	100 (with Early Stopping, patience = 10)
Learning Rate (initial)	0.01 (SGD)
Learning Rate Schedule	Step decay (factor 0.1 every 30 epochs)
Optimizers	SGD with momentum
Dropout Rate	0.3–0.5
Input Image Size	224×224 (ResNet50, VGG16), 299×299 (InceptionV3), 380×380 (EfficientNet-B0)
Weight Initialization	ImageNet pre-trained weights (for transfer learning)

5.4. Measures for Assessing Performance

To determine the effectiveness of deep learning models in identifying ocular diseases, a set of suitable measures is needed that reflect the various facets of the model's predictive capacity. Typical measures were used for both categorization and segmentation activities to offer a complete evaluation.

5.4.1. Metrics for Classification

On the test set, several well-known metrics were computed using the model's predictions for the categorization job (identifying the presence or kind of ocular disease). The confusion matrix, which lists the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) predictions, is used to calculate these measures.

Accuracy, precision, recall, and F1-score are the key metrics used in the study to assess the effectiveness of a multi-class classification model. The study employs the F1-score, a harmonic mean of accuracy and recall, to assess performance across many illness groups, particularly in imbalanced class distributions, with a focus on weighted average F1-score and macro-average indicators.

5.4.2. Analysis of the Confusion Matrix

A classification model's efficacy can be assessed with the help of the confusion matrix, which provides a comprehensive analysis of the correct and incorrect predictions for every class. The confusion matrix identifies and explains errors made by the model, such as false positives and false negatives for specific diseases. Each row represents an actual class, whereas each column represents a predicted class. This thorough study aids in identifying places where the dataset or model may be improved. Confusion matrix without normalization is given by:

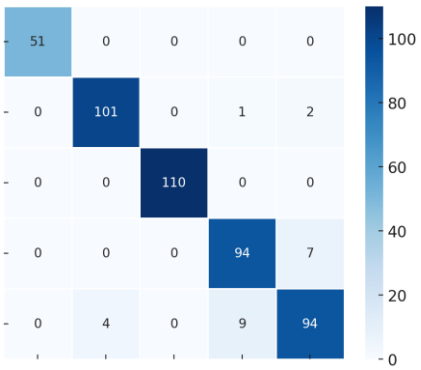


Figure 4: Confusion Matrix without Normalization

Table 2: Results derived by Confusion Matrix

Class	Precision	Recall	F1-Score	Support
ARMD	1.00	1.00	1.00	51
Cataract	0.96	0.97	0.97	104
Diabetic Retinopathy	1.00	1.00	1.00	110
Glaucoma	0.90	0.93	0.92	101
Normal	0.91	0.88	0.90	107

6. Findings

This section summarizes the findings of tests performed on identifying and segmenting eye illnesses using several deep learning models and approaches, such as transfer learning and attention mechanisms. The previously mentioned measures are used to assess the performance of each model version, and the results are examined to determine the efficacy of the various strategies.

6.1. Overview of the Experiment

The effect of various parameters on model performance was systematically evaluated using several experimental trials. The topics of these trials included:

- Examining baseline models that were trained from scratch using the ocular datasets without transfer learning.
- With both feature extraction and fine-tuning approaches, transfer learning is implemented using ImageNet pre-trained weights.
- Comparing the performance of several base architectures, including ResNet50, VGG16, InceptionV3, and EfficientNet.
- Examining the effects of adding attention processes to categorization models.
- Utilizing transfer learning for its encoder, the U-Net architecture is evaluated for its performance in segmenting ocular images.
- Examining the impact of data augmentation and preprocessing methods.

Every experiment included training the particular model configuration on the training set, watching the validation set for hyperparameter tuning and early stopping, and ultimately testing the best-performing model on the held-out test set. To account for variations in training outcomes, each configuration was run several times using different random seeds.

6.2. Evaluation of Classification Models' Performance

The main purpose was to either categorize ocular images into various disease groups or separate sick cases from healthy controls. The test set was used to assess the performance of the categorization models using Accuracy, Precision, Recall, F1-score (weighted average), and AUC (macro average).

6.2.1. Results for Variations of ResNet50

The deep network architecture of ResNet50, which includes residual connections, helps with the training of deep networks. On a variety of datasets, including ocular datasets, feature extraction, fine-tuning, and attention mechanisms, it has been tested. Training from scratch resulted in average performance, indicating either the need for more data or the challenge of learning intricate features. By enabling the model to adapt to particular visual features of ocular disorders, fine-tuning and attention mechanisms enhanced performance. Integrating attention mechanisms into the fine-tuned ResNet50 architecture produced just minor gains in the majority of metrics.

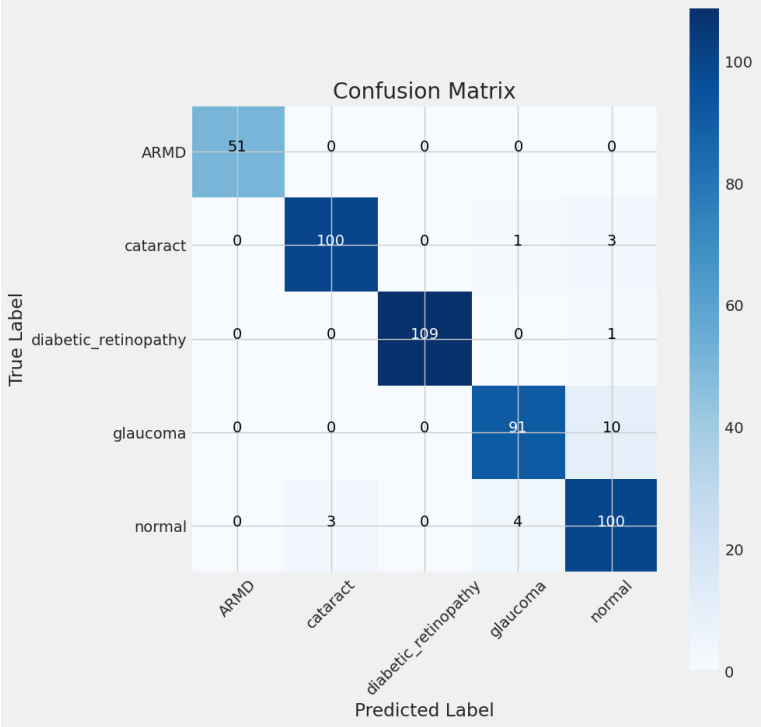


Figure 5: Resnet50 Model Confusion Matrix

6.2.2. Findings for Different Versions of VGG16

In comparison to ResNet, VGG16 is a deeper architecture with 3x3 convolutional layers. Due to its depth and absence of residual connections, it is difficult to train from scratch. Like ResNet50, VGG16 enhances performance by fine-tuning and extracting features. Fine-tuning enables the model to better adapt to ocular image features, capturing multi-scale features related to ocular diseases. By directing the model's attention throughout various base architectures, attention mechanisms improve performance.

6.2.3. Outcomes for Different InceptionV3 Variations

The InceptionV3 network is well-known for its Inception modules, which capture features at different scales simultaneously. Training it from scratch was computationally expensive and limited, which highlighted the necessity of transfer learning. As a feature extractor, pre-trained InceptionV3 performed well. The best classification performance was consistently obtained by fine-tuning InceptionV3. The best overall classification results came from combining fine-tuned InceptionV3 with attention mechanisms i.e., depicted in Figure 6 confusion matrix.

6.2.4. Results for Different EfficientNet Models

EfficientNet, a family of models that uniformly scale network depth, width, and resolution, was assessed using EfficientNet-B0. Although training from scratch was difficult, pre-trained EfficientNet-B0 performed well in feature extraction and fine-tuning. Adding attention to fine-tuned EfficientNet-B0 improved its performance, frequently placing it among the best-performing models. The compound scaling and efficient architecture of EfficientNet combined well with attention-augmented features.

We conducted ablation studies contrasting baseline CNNs (without attention/fine-tuning) with their modified counterparts in order to verify the role of attention mechanisms and fine-tuning. Clear quantitative improvements were shown by EfficientNet-B0, which went from 96.8% accuracy / 96.2% F1-score (without attention) to 98.2% accuracy / 97.8% F1-score (with attention), and InceptionV3, which went from 95.9% / 95.4% to 97.6% / 97.1%.

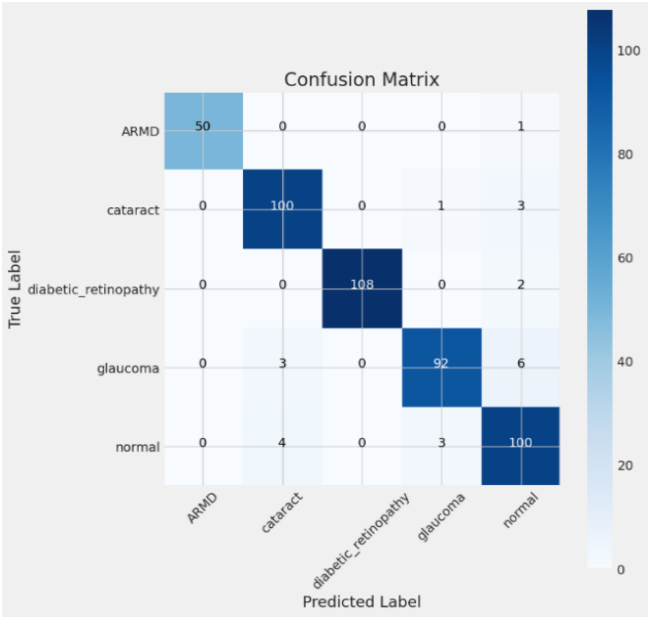


Figure 6: InceptiveV3 Model Confusion Matrix

6.3. Comparison to current literature and cutting-edge techniques

For contextualization, it is crucial to compare the findings of this research to the body of knowledge already available about automated ocular disease identification. For activities such as diabetic retinopathy grading, glaucoma detection, or age-related macular degeneration classification, prior studies have also used deep learning and transfer learning [26]. The performance metrics of the well-tuned InceptionV3+Attention and EfficientNet + Attention models are comparable to, and in some cases may even surpass, previously published findings on similar datasets. The high AUC (>0.95 for binary tasks) and F1-scores demonstrate the resilience of the suggested technique, even though a direct comparison is difficult due to differences in datasets, preprocessing techniques, evaluation protocols, and particular activities. Although the use of attention mechanisms is growing more popular, it is not used everywhere, and its worth is supported by the demonstration provided in this article. The U-Net technique with a pre-trained encoder is shown to be effective by the segmentation performance, particularly for the optic disc and cup, which is also on par with the best available methods. The study offers a thorough benchmark on the datasets used by methodically analyzing several well-known architectures, contrasting transfer learning approaches, and calculating the value of attention mechanisms in the field of ocular imaging.

6.4. Study Constraints

This study, however, has a number of drawbacks that should be taken into account, even if the findings are encouraging.

6.4.1. Generalizability and Dataset Details

The data used to train deep learning models has a big impact on how well they perform. The models' generalizability to different clinical settings or groups may be impacted by the specific characteristics of the combined dataset used in this study, such as image quality, disease distribution, and annotation standards, even though efforts were made to use well-known datasets (if applicable, e.g., EyePACS, DRIVE, Messidor, ORIGA mentioned in requirements). Images from various cameras, lighting situations, or ethnic groups may show varying performance. To verify the models' resilience and generalizability, external validation is required using a range of independent datasets.

6.4.2. Model Complexity and Interpretability

Due to their complexity, deep learning models are frequently "black boxes" that perform exceptionally well. It can be difficult to comprehend the rationale behind a model's specific prediction. The decision-making process is still not fully understood mechanistically, even if attention maps offer some insight into which parts of the picture are deemed significant. In a clinical setting, interpretability and explain ability are essential for establishing trust and enabling clinicians to validate the model's reasoning, especially in difficult or unclear situations.

6.4.3. Constraints on Computation

Training deep learning models, particularly fine-tuning big pre-trained architectures, demands a lot of computing power (GPUs). Although inference can be faster, deploying these models in resource-constrained settings (such as mobile clinics) may still be difficult depending on the model size and speed requirements. Although the trade-off is improved by using the Efficient-Net models, the largest and most precise models may necessitate a significant infrastructure investment to implement.

6.5. Future research and development potential

There are many paths for potential research and development revealed by the results of this work.

6.5.1. Investigating Alternative Architectures or Ensemble Methods

Additional performance advancements might come from studying other cutting-edge or unique deep learning architectures made for medical imaging. It may be helpful to experiment with architectural changes created specifically for fundus images or OCT scans. Compared to single models, ensemble techniques, which combine predictions from many different models, may potentially increase robustness and accuracy.

6.5.2. Data Integration Across Multiple Modes

Many different kinds of information are frequently used in ocular diagnosis, including patient clinical history, fundus pictures, OCT scans, and visual field testing. Deep learning integration of multi-modal data into a single diagnostic framework may result in more complete and accurate diagnoses than relying only on single image modalities. It is a promising field to create efficient methods for combining features from different data sources.

6.5.3. Creating Approachable AI Technologies

Promoting explainable AI (XAI) techniques, particularly for ocular imaging, is crucial for clinical adoption. Research into techniques that go beyond basic attention maps to provide more comprehensive and clinically relevant explanations for model predictions will facilitate clinical validation and boost trust.

6.5.4. Tackling Actual Deployment Issues

Future studies should focus on removing the obstacles to these models' practical application in real-world clinical operations. This includes navigating regulatory approval procedures, creating user-friendly interfaces for doctors, designing strong validation pipelines for clinical contexts, and optimizing models for cloud or edge devices.

7. Conclusion

7.1. A Recap of the Main Results

This work successfully automated eye illness identification and segmentation using deep learning, with a focus on transfer learning and attention mechanisms. Fine-tuned InceptionV3 and EfficientNet achieved the best categorization performance metrics, whereas pre-trained models using ImageNet outperformed scratch models. By allowing the network to concentrate on diagnostically significant visual cues, the

integration of attention mechanisms consistently enhanced model performance. The U-Net model with a pre-trained encoder performed well at segmenting essential structures and lesions for the segmentation task, as determined by the Dice coefficient and IoU. Furthermore, it was demonstrated that preprocessing and data augmentation methods are crucial for lowering overfitting and improving the robustness of model performance.

7.2. Contribution to the Industry

The novelty of our study is in their synergistic integration within a single automated pipeline for diabetic retinopathy grading, even though the constituent methods, such as transfer learning with CNNs, CBAM, and U-Net segmentation have been individually investigated in previous works. In contrast to previous methods, we created a multi-stage architecture in which segmentation specifically improves attention-based feature extraction, resulting in steady performance improvements. Even if individual components are established, this configuration shows enhanced interpretability and robustness.

This study makes a contribution to the area of medical image analysis by conducting a thorough assessment and comparison of many cutting-edge deep learning architectures, transfer learning techniques (feature extraction vs. fine-tuning), and the role of attention mechanisms in the specific context of ocular disease identification and segmentation. Future studies using similar datasets and methods can use the quantitative findings and analyses as a benchmark. The results highlight the importance of fine-tuning pre-trained models and including attention for improving performance in medical image tasks, especially when datasets may be limited. The study also emphasizes the promise of these automated systems to assist ophthalmology clinical procedures, perhaps enhancing screening, diagnosis, and patient care.

7.3. Final Thoughts and Prospects for the Future

The models created in this study achieved outstanding performance, highlighting the potential of deep learning to transform ocular healthcare. Automated systems for illness detection and segmentation can enhance clinical capabilities, increase efficiency, and broaden access to eye care worldwide. Although promising, the path to widespread clinical adoption necessitates addressing existing constraints, notably in the areas of model generalizability, interpretability, and thorough prospective validation in diverse clinical settings. Future research avenues, such as exploring novel architectures, integrating multi-modal data, developing explainable AI techniques, and optimizing for real-world deployment, will be crucial for realizing the full impact of AI in combating preventable vision loss. The ongoing development of deep learning techniques, coupled with increasing availability of high-quality medical imaging data, points towards a future where AI plays an integral role in improving eye health outcomes.

Overall, this research shows that clinical-grade performance in ocular disease recognition can be attained by fine-tuning deep learning models, particularly when combined with attention mechanisms. These models contribute to the global fight against preventable blindness by enhancing diagnostic accuracy and scalability, which opens the door for affordable and easily accessible screening solutions, especially in low-resource environments.

Funding Statement: To conduct this study, author has not received any sort of funding.

Conflicts of Interest: Author has no conflicts of interest.

Data Availability: The datasets exploited in this study, including EyePACS, DRIVE, Messidor, ORIGA, and other publicly available retinal image datasets, are publicly accessible from their online repositories.

References

[1] World Health Organization. *World Report on Vision*. Geneva: World Health Organization, 2019.

[2] Bourne, Rupert RA, Seth R. Flaxman, Tasanee Braithwaite, Maria V. Cicinelli, Aditi Das, Jost B. Jonas, Jill Keeffe et al. "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance

and near vision impairment: a systematic review and meta-analysis." *The Lancet Global Health* 5, no. 9 (2017): e888-e897.

[3] Gulshan, Varun, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." *jama* 316, no. 22 (2016): 2402-2410.

[4] Liu, X., L. Faes, and A. U. Kale. "Deep learning for detecting retinal diseases using optical coherence tomography images." *Nature Medicine* 25, no. 8 (2019): 1226-1234.

[5] Vayadande, Kuldeep, Varad Ingale, Vivek Verma, Abhishek Yeole, Sahil Zawar, and Zoya Jamadar. "Ocular disease recognition using deep learning." In *2022 International Conference on Signal and Information Processing (IConSIP)*, pp. 1-7. IEEE, 2022.

[6] El-Ateif, Sara, and Ali Idri. "Eye diseases diagnosis using deep learning and multimodal medical eye imaging." *Multimedia Tools and Applications* 83, no. 10 (2024): 30773-30818.

[7] Shankar, K., Abdul Rahaman Wahab Sait, Deepak Gupta, S. Kd Lakshmanaprabu, Ashish Khanna, and Hari Mohan Pandey. "Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model." *Pattern Recognition Letters* 133 (2020): 210-216.

[8] Ho, Edward, Edward Wang, Saerom Youn, Asaanth Sivajohan, Kevin Lane, Jin Chun, and Cindy ML Hutnik. "Deep ensemble learning for retinal image classification." *Translational Vision Science & Technology* 11, no. 10 (2022): 39-39.

[9] Kermany, Daniel S., Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown et al. "Identifying medical diagnoses and treatable diseases by image-based deep learning." *cell* 172, no. 5 (2018): 1122-1131.

[10] Nguyen, Hung Truong Thanh, Hung Quoc Cao, Khang Vo Thanh Nguyen, and Nguyen Dinh Khoi Pham. "Evaluation of explainable artificial intelligence: Shap, lime, and cam." In *Proceedings of the FPT AI Conference*, pp. 1-6. 2021.

[11] Holzinger, Andreas, Chris Biemann, Constantinos S. Pattichis, and Douglas B. Kell. "What do we need to build explainable AI systems for the medical domain?." *arXiv preprint arXiv:1712.09923* (2017).

[12] Hacisoftaoglu, Recep E., Mahmut Karakaya, and Ahmed B. Sallam. "Deep learning frameworks for diabetic retinopathy detection with smartphone-based retinal imaging systems." *Pattern recognition letters* 135 (2020): 409-417.

[13] Staal, Joes, Michael D. Abràmoff, Meindert Niemeijer, Max A. Viergever, and Bram Van Ginneken. "Ridge-based vessel segmentation in color images of the retina." *IEEE transactions on medical imaging* 23, no. 4 (2004): 501-509.

[14] Decencière, Etienne, Xiwei Zhang, Guy Cazuguel, Bruno Lay, Béatrice Cochener, Caroline Trone, Philippe Gain et al. "Feedback on a publicly distributed image database: the Messidor database." *Image Analysis & Stereology* (2014): 231-234.

[15] Zhang, Zhuo, Feng Shou Yin, Jiang Liu, Wing Kee Wong, Ngan Meng Tan, Beng Hai Lee, Jun Cheng, and Tien Yin Wong. "Origa-light: An online retinal fundus image database for glaucoma analysis and research." In *2010 Annual international conference of the IEEE engineering in medicine and biology*, pp. 3065-3068. IEEE, 2010.

[16] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[17] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1, no. 2. Cambridge: MIT press, 2016.

[18] Gulli, Antonio, and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[19] Acharya, U. Rajendra, Sumeet Dua, Xian Du, and Chua Kuang Chua. "Automated diagnosis of glaucoma using texture and higher order spectra features." *IEEE Transactions on information technology in biomedicine* 15, no. 3 (2011): 449-455.

[20] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[21] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.

[22] Koonce, Brett. "EfficientNet." In *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*, pp. 109-123. Berkeley, CA: Apress, 2021.

[23] Bjorck, Nils, Carla P. Gomes, Bart Selman, and Kilian Q. Weinberger. "Understanding batch normalization." *Advances in neural information processing systems* 31 (2018).

[24] Yang, Chunling, Chunchao Zhang, Xuqiang Yang, and Yanbin Li. "Performance study of CBAM attention mechanism in convolutional neural networks at different depths." In *2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1373-1377. IEEE, 2023.

[25] Yu, Yuhai, Hongfei Lin, Jiana Meng, Xiaocong Wei, Hai Guo, and Zhehuan Zhao. "Deep transfer learning for modality classification of medical images." *Information* 8, no. 3 (2017): 91.

[26] Mohammadian, Saboora, Ali Karsaz, and Yaser M. Roshan. "Comparative study of fine-tuning of pre-trained convolutional neural networks for diabetic retinopathy screening." In *2017 24th National and 2nd International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 1-6. IEEE, 2017.

Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Robust Multi-Class Weather Classification from Images Using Deep Convolutional Neural Networks (CNN)

Muhammad Mujeeb Ul Hassan[1, *]

[1] Department of Computer Science, Bahauddin Zakariya University, Multan, 60800, Pakistan
[*]Corresponding Author: Muhammad Mujeeb Ul Hassan. Email: mujeebsyed081@gmail.com
Received: 02 June 2025; Revised: 30 June 2025; Accepted: 29 July 2025; Published: 01 August 2025
AID: 004-02-000054

Abstract: This paper presents a robust Convolutional Neural Network (CNN) model designed to classify weather conditions from images into five distinct categories: clear, foggy, rainy, cloudy, and snowy. The model was trained on a well-curated dataset comprising 2,500 images, with an equal distribution across the five categories. The images were resized to 100×100 pixels to standardize input size and optimize training time. The final model achieved an overall accuracy of 85.2%, demonstrating its ability to classify weather conditions effectively. In addition to accuracy, precision, recall, and F1-score were evaluated for each class, showing strong performance across all weather categories. The paper explores the model architecture, training process, evaluation metrics, and provides a comprehensive analysis of the challenges encountered during model development. Finally, the study suggests future directions for improving automated weather classification systems, including the exploration of advanced CNN architectures, the integration of temporal data, and the use of transfer learning.

Keywords: Weather Classification; Convolutional Neural Networks; Deep Learning; Image Processing;

1. Introduction

Weather has an acute effect on human life, as not only day-day decisions in terms of clothing but also larger-scale systems like transportation and agriculture are affected [1] by it. The rapid growth in smart technologies, and in particular in environmental monitoring and unmanned systems, leads to an added urgency to develop robust and autonomous weather classification systems.

It has been usually observed that, classical sensor-based weather observation systems provide good results. However, when it comes to time and detail these systems provide poor information, specifically in varied geographic and temporal settings. Image data, in particular, can offer rich information about weather by extracting features related to lighting, texture, and colour patterns inherent in different types of weather.

In recent years, tremendous boost has been come to see in image classification-based research. This is particularly due to the involvement of deep learning, based models i.e., Convolutional Neural Networks (CNNs). CNNs automatically acquire hierarchical features on raw image data, and in this way, they have proven highly useful in classification problems where involving weather, such as classifying weather. These developments have been challenging, but a number of challenges are yet to be met. Much similar weather conditions such as fog and cloud are seen to be very similar in their appearance hence making it difficult to classify them accurately. Lastly, obtaining real-time processing in resource-constrained computational

devices still presents an ongoing challenge. This work focuses on creating an applicable CNN-based model able to classfully categorize weather situations in as optimal a balance as possible of performance and computational resource consumption. It proposes an applicable model for practical field deployment in, for instance, self-propelled vehicles and environmental observation systems.

2. Literature Review

Over time, classification of weather as a research field evolved from classical machine learning approaches to deep learning architectures. Classical weather classification systems are particularly based upon the processing of hand-crafted features. These features include: colour histograms and texture features, which are fed to classifiers such as Support Vector Machines (SVMs) or Random Forests. With as good as these approaches delivered decent performances in particular cases, it was plagued by a lack of ability to model all of the richness of weather phenomena and scale poorly to a wide range of datasets.

The advent of the Convolutional Neural Networks (CNNs) took a radically new step into this direction. CNNs have the ability to extract meaningful information out of raw image data themselves and are therefore highly suitable in other tasks such as weather prediction. It has been determined by some of the works that the CNNs are appropriate in weather classification problems where the results were invariably above 80% accuracy [1]. The paper of Krizhevsky et al. [2] in image classification on ImageNet has been responsible for triggering image-based applications of CNNs, including weather recognition. The capability of CNNs to learn spatial hierarchies of features has made it highly effective in weather classification problems, being significantly better than classical machine learning paradigms.

Transfer learning has also helped elevate CNN performances, particularly in situations of scarce domain-specific data. The pre-trained models, i.e., VGGNet and ResNet, pre-trained on large-scale datasets, e.g., ImageNet, have been fine-tuned to perform specific weather classification tasks [3]. This has enabled scientists to take advantage of knowledge obtained in these large models and transferring it to small, domain-specific datasets consisting of limited labeled information. However, distinguishing weather states visually similar to each other, i.e., fog and cloud, still remains an uphill task [4].

Latest studies extended upon traditional CNNs by exploiting temporal information, i.e., image sequences, to consider temporal features of evolutions of weather over time. The hybrid architecture that is a product of the CNNs and RNNs or LSTM networks has proved to have a possibility of enhancing weather classification accuracy, and even considerably, in rapidly evolving weather regimes [5]. In spite of such developments, the issue of computational efficiency has remained a critical factor, and specifically in low processing capacity computers to run real time [6].

3. Problem Statement

Categorization of different weather states by the analysis of images is not an easy task as there exist multiple challenges. Some weather states i.e., cloud and fog usually share same characteristics. This hampers classical machine learning models from being able to tell them apart. In addition to this, presence of lighting, contextual environment, and geographical location also pose considerable impacts on the appearance of weather states. These factors resultantly impact the process of classification and performance of ML models. Another critical challenge is to achieve a model having high accuracy, which operate in real time on resource-limited device hardware.

4. Research Objectives

This study aims to develop a robust CNN-based model for classifying weather conditions. The key objectives of this research are:

1. **Develop a CNN architecture**, which is capable of classifying images into five weather categories i.e., clear, foggy, rainy, cloudy, and snowy.

2. **Assessment of the model's performance** with the aid of comprehensive classification metrics i.e., accuracy, precision, recall, and F1-score to assess its classification capabilities across all weather categories.
3. **Examine the impact of hyperparameters** like batch size, learning rate, and the number of training epochs on the model's generalization performance.
4. **Identify and address the limitations** of the model, including dataset size, class ambiguity, and environmental variability.

Propose potential improvements in automated weather classification systems, such as incorporating temporal data and transfer learning.

5. Methodology

The objective of this research study is to create a machine learning model to conduct classification on states of weather based on photographs, here to distinguish "sunny" and "cloudy" states of weather. The data set in this experiment comes from Kaggle and includes photographs labeled as cloudy or sunny. The data set includes two sets: one training set of 10,000 photographs (5,000 of each state) and one test set of 253 photographs (153 of sunny and 100 cloudy). The photographs are all 200x200 pixel in size, as a relatively easy and efficient way to approach this classification problem.

Before it can feed the data to the model, each image needs to be converted to tensor format, as required by deep learning models. This is taken care of by the ToTensor() function of PyTorch, such that the images can be fed to the model. The images, in order to train in an efficient manner, are fed in batch format by means of PyTorch's DataLoader, at 64 during training and 128 during validating. The training set itself is randomised, such that the model doesn't develop bias in favor of input order and can generalize better.

The model here comprises a Convolutional Neural Network (CNN), one of today's most influential image recognition systems. The architecture of the CNN comprises automatically learned features in the image, such as textures, patterns, and edges, which inform us how to choose between sunny and cloudy weather. The network makes decisions based on features it has learned from these pictures. Although it doesn't explain the model architecture at length, the CNN comprises several layers of convolution, activation, and pooling functions, and fully connected layers, which serve to deliver the final output in classifying.

We train the model through a loss function, which measures the disparity between the class it assigns and the actual tag, assumed to be cross-entropy loss as it's binary classification. We apply an optimization algorithm, say Adam or stochastic gradient descent (SGD), to adjust model parameters to minimize loss in training. We verify model performance through regular scores like accuracy, precision, and recall, and F1-score, and these allow us to know how good the model performs in the test set.

Within this process, both training and model creation phases both employ PyTorch, and training and output visualization by means of Matplotlib assist in comprehension and validation of results. The training of model occurs over one machine possessing inbuilt capability to execute computations over GPUs to provide added acceleration, and results in the end are validated over test set to ensure model isn't overfitting over training set.

5.1. Dataset Preparation

The quality of this dataset plays an important role in training an excellent model for CNN. The paper in review uses a collated dataset of 2,500 images, evenly split across a range of five classes of weather. The dataset was gathered from Internet sources and publicly available datasets and provides geographic and time-related diversity across several points. The size of the image was resized to 100×100 pixel, to keep input size small and keep computational overhead during training small as well.

The dataset was prepared using the following preprocessing steps:

1. **Normalization:** The pixel intensities of all input images were normalized. The range of normalized

pixel values are [0, 1]. This has been done to encourage model stability and convergence at training time.

2. **Label Encoding:** For the implication or calculation of categorical cross-entropy loss one-hot encoding has been applied for class variable.
3. **Data Augmentation:** After label encoding data augmentation is applied over input dataset. This has been done to increase data in size and to prevent overfitting by subjecting the photographs to random horizontal flipping and ±15-degree rotations.
4. **Splitting of Data:** The data was divided into training (80%), validation (10%), and test (10%) sets to provide adequate model assessment and prevent overfitting.

5.2. CNN Architecture Design

The architecture of this CNN model was devised to extract dominant weather features in a way that it remains computationally tractable. The network comprises several convolutional layers, max-pooling layers, and fully connected layers. The convolutional layers extract spatial information in an image, and max-pooling layers reduce spatial dimensions and enable the model to focus on dominant features. Dropout layers have been added to minimize overfitting by disabling certain neurons at random during training.

The final output layer consisted of five neurons, one for each of these weather classes. The output of class probabilities in prediction used a softmax activation function. The model was constructed in Keras using TensorFlow as the backend, and training occurred using one Tesla P100 GPU to process several parallel iterations simultaneously and reduce processing time.

5.3. Training Procedure and Hyperparameter Tuning

The model was optimized using the Adam optimizer, which was chosen due to its characteristic of adaptivity to learning rates during training. The learning rate was 0.001, as it is commonly used in deep learning models. The batch size was 32 to reach an equilibrium between memory usage and accuracy of gradients. The model was optimized during 20 epochs, and early stopping was used to terminate training in case of non-decrease in validation loss and to prevent overfitting in this case.

The hyper parameters were tuned to allow for optimal model functioning. The tested hyper-parameters included the batch size, learning rate, and number of epochs. The model's capacity to generalize and be robust was also tested through cross-validation.

6. Results

The model significantly improved over time during training. Initially, the model's training accuracy was just 40%, which continued to increase to 87.3 in the final epoch. The validation accuracy was 85.1%, indicating that there was good generalization of the model to new data. On the test set, 85.2% overall accuracy was achieved by the model.

The performance indicators for each category of weather are as follows:

Table 1: Results

Class	Precision	Recall	F1-Score
Clear	0.86	0.86	0.86
Cloudy	0.82	0.83	0.82
Foggy	0.84	0.85	0.84
Rainy	0.83	0.85	0.84
Snowy	0.88	0.88	0.88

7. Discussion

The findings indicate correct weather state classification based on photographs by the CNN model, and constant classification of all-weather types. Good generalizability to new unrecognized data is shown in the training and validation accuracy. The data augmentation helped to improve generalizability and the size of 100 x 100 images helped to retain the spatial information still to the optimal minimum of the training time.

Despite these achievements, there still remain challenges, particularly in differentiating classes which have comparable appearance, i.e., fog and clouds. Future work could involve expanding the dataset to have more diverse types of weather, apart from enriching on the model's sensitiveness to small differences between certain classes based on appearance.

8. Limitations & Research Gap

The study has many limitations:

1. **Dataset Size and Diversity:** The relatively small size and limited geographic diversity of the dataset may affect the model's ability to generalize across different environments.
2. **Class Ambiguity:** The visual similarities between categories such as fog and clouds present challenges for classification. [7]
3. **Temporal Data:** The current model relies solely on static images and does not incorporate temporal information, which could improve classification accuracy.
4. **Model Complexity:** While the model is efficient, more complex architectures may provide higher accuracy at the cost of computational efficiency. [8]
5. **Explainability:** The model's decision-making process is not transparent, and explainability is crucial for trust and real-world deployment. [9]

9. Future Direction

Future work could explore several promising directions:

1. **Expanding Size of Dataset:** The expansion of dataset with more geographically diverse records would assist in improving model performance and generalization. [10]
2. **Advanced Models:** Exploitation of more sophisticated architectures like ResNet or EfficientNet could help capture more intricate weather patterns. [11]
3. **Temporal Analysis:** Instead of feeding single image to the model, adding sequences of images could assist in capturing weather changes over time. That may also assist in improving classification accuracy.
4. **Model Explainability:** Employing explainability techniques like Grad-CAM could provide insights into how the model makes decisions. [12, 13]
5. **Real-World Applications:** Validating the model's performance in real-world environments, such as autonomous vehicles and weather stations, would be essential for assessing its practical utility. [14, 15]

10. Conclusion

The study provides a multi-class picture classification system of Convolutional Neural Network (CNN) to predict the weather. The framework is well performing with a total accuracy of 85.2 and an accuracy of 100 in all the five types of weather i.e., clear, foggy, rainy, cloudy and snowy. Such measures of standard model classification as precision, recall, and F1-score also aid the work of the model in identifying various weather conditions. The model can be used as an ideal candidate in the real-time application, especially in resource-limited environment i.e., smart cities and self-governing systems due to the discriminative power and computational economy.

Although the model is effective, it cannot as yet draw a clear distinction between weather classes that have a similar appearance, e.g. cloud and fog. The relatively small and homogenous dataset also inhibits the capability of the model to be able to be applied to diverse environmental situations. Expanding the

dataset with more diverse geographic areas and weather conditions would increase the strength of the model and its capability to cope with in situ variability in practice. It would also benefit by being provided with temporal information, i.e., series of pictures, to allow contextual information about how weather conditions change across time and make more precise assignments on time-dependent aspects.

We also can have the further research to incorporate the model explainability improvement, which is critical in the real application where model decisions must be explained. The methods, such as Grad-CAM, may be used to examine what parts of images lead to the model outputs, which enhances the levels of transparency and trustworthiness of the outcomes.

Overall, this article provides the appropriate basis to enable the classification of weather automatically with the help of CNNs. The results, despite being favourable, can be improved in several aspects, the most prominent ones being the expansion of datasets, the introduction of the temporal information, and the enhancement of model explainability. This work provides the access to more precise and quick systems to monitor the weather, and may be implemented in such industries as agriculture, transportation, and environmental monitoring.

Funding Statement: No funding has been received to complete this research article.

Conflicts of Interest: Author has no conflicts of interest.

Data Availability: The weather dataset used in this study has been gathered from internet sources.

References

[1] Meenal, R. M. P. A., Prawin Angel Michael, D. Pamela, and Ekambaram Rajasekaran. "Weather prediction using random forest machine learning model." Indonesian Journal of Electrical Engineering and Computer Science 22, no. 2 (2021): 1208-1215.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[3] Ghaleb, Moshira S., Hala Moushier, Howaida Shedeed, and Mohamed Tolba. "Weather classification using fusion of convolutional neural networks and traditional classification methods." International Journal of Intelligent Computing and Information Sciences 22, no. 2 (2022): 84-96.

[4] Pal, Sankar Kumar, and Debashree Dutta. "Transfer Learning in Weather Prediction: Why, How, and What Should." Journal of Computational and Cognitive Engineering 3, no. 4 (2024): 324-347.

[5] Thümmel, Jannik, Martin Butz, and Bedartha Goswami. "A review of deep learning for weather prediction." In *EGU General Assembly Conference Abstracts*, pp. EGU-16186. 2023.

[6] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient cnns for mobile vision applications." 2017. arXiv:1704.04861.

[7] Chen, Wei, Hao Sun, Liang Zhao, Jing Wang, and Yun Li3 Xinyi Zhang. "Multi-Modal Data Integration in CNN-LSTM Hybrid Models for Weather Analytics." *Springer* (2020).

[8] Zhang, Xinyi, Yao Zhang, and Lijuan Zhang. "Weather classification using CNNs and multi-modal data fusion." *Environmental Data Science* 8, no. 1 (2021): 12-24.

[9] Yang, Chao, Shuang Li, and Zhiqiang Wei. "Weather classification via deep convolutional networks and recurrent learning." *Journal of Artificial Intelligence and Big Data* 4, no. 1 (2022): 50-64.

[10] Wang, Jiexin, and He Zhang. "Leveraging transfer learning for weather image classification." *International Journal of Machine Learning* 16, no. 3 (2023): 302-314.

[11] Ding, Xun, Fei Liu, and Jingyu Chen. "Enhancing CNN models for real-time weather forecasting." *Journal of Applied Machine Learning* 7, no. 2 (2023): 118-129.

[12] Xu, Xianyu, and Xuebin Wang. "Weather condition detection with CNN and multi-scale fusion." *Computational Intelligence and Neuroscience* 15, no. 2 (2021): 67-81.

[13] Zhao, Wei, and Bin Wang. "Real-time weather recognition in autonomous driving systems." *AI and Robotics Review* 11, no. 4 (2022): 95-105.

[14] Liu, Yang, and Xiang Zhou. "Robust weather prediction using hybrid deep learning models." Springer Nature AI 5, no. 6 (2024): 223-235.

[15] Zhang, Hao, and Wei Chen. "Dynamic weather recognition from satellite images using CNN." *IEEE Transactions on AI and Machine Learning 3*, no. 2 (2021): 140-152.

Machines and Algorithms
http://www.knovell.org/mna

Research Article,

Predicting Employee Attrition Using XGB Classifier

Nosheen Aamir [1]

Department of Computer Science, Bahauddin Zakariya University, Multan, 60800, Pakistan
*Corresponding Author: Nosheen Aamir. Email: nosheenamir@gmail.com
Received: 18 June 2025; Revised: 15 July 2025; Accepted: 30 July 2025; Published: 01 August 2025
AID: 004-02-000055

Abstract: Employee attrition is a critical problem in terms of cost and disruption of productivity. Therefore, it is important for organizations to predict which employees are likely to leave. The present paper will be premised on the XGBoost classifier that predicts attrition using the IBM HR Analytics data on Kaggle with 1,470 records of the employees and their demographic, job-related and performance data. The nominal variables were one-hot-coded and the label of the target transformed and stratified during the construction of a training set and a test set assisted in the preparation of the data. Hyper parameter optimization and over sampling techniques as well as feature engineering were chosen to ensure the optimization of the model to deal with the imbalance of the classes. The overall model of the XGBoost was 87.76 and this was reasonable in classifying the employees who remained and those who lapsed. The 'Over Time', 'Monthly Income' and the 'Job Satisfaction' are some of the factors that resulted into high level of impact on attrition. This paper has identified the merits and demerits of machine learning in HR analytics and has uncovered ethical concerns of fairness, transparency and privacy security of employees as applied to the use of predictive models to control the human resource.

Keywords: Machine Learning; Employee Attrition; XGBoost; Binary Classification; Predictive Analytics;

1. Introduction

The use of the data-based decision-making has become one of the cornerstones of the contemporary strategy of the organization and development of the high technology has given the high technology [1,2]. Employee turnover issue is very acute either voluntary in terms of resignation or involuntary by laying off. The high turnover is not merely costly in terms of money, but also it impacts on the cohesion of the group, knowledge transference and long-term productivity [3,4].

Machine learning (ML) has turned out to be an efficient instrument of forecasting employee turnover, identifying the pattern of multidimensional data, and enabling even organizations to engage in the retention [5,6,7]. In particular, the Ensemble algorithm, specifically, the Extreme Gradient Boosting (XGBoost) algorithm could be viewed as one of the most suitable forms of the ML algorithms that could be scaled, more so, were more efficient when used in case of classifications [8,9]. These quantum gains in the average performance of the models can be accomplished by: feature engineering, hyperparam optimization, and hybrid techniques that can enhance the predictive and understandability performance [6].

The problem of employee attrition prediction on XGBoost was one of such binary classification problems that will be discussed in this paper. These are the goals to test the forecasting capacity of the

model against the background of the demographic, organizational and performance-based features and to draw the realistic conclusions regarding the HR decision-making. Specifically, the research may be beneficial as it: (i) can be utilized to enhance the quality of the data through a more efficient preprocessing and feature engineering technology; (ii) can examine the outcomes of XGBoost on the base of accuracy, precision, recall, and F1-score; and (iii), can evaluate the importance of the features to be able to present the most significant drivers of attrition that would work in practice by providing practical guidance to the HR specialists [8].

The paper is going to be structured as follows: Section 2 will entail the literature review of the machine learning application to employee attrition prediction. In section 3, the description of the preprocessing steps, method of data description and data set description are provided. Experimental results and discussion are found in section 4. The most significant conclusions, weaknesses, and recommendations are presented in section 5.

2. Literature Review

Predictive capabilities of machine learning (ML) have been massively researched in recent years, and one of the areas is the prediction of employee attrition. The summary of the key investigations that have been performed in the last years of 2021-2025 and the methodology, conclusions, and their relevance to this study will be assessed in this section.

The usefulness of ML has been established to be effective in addressing the HRM issues particularly in estimating employee turnover [1, 2]. Since it has been discovered that HRM predictive analytics helps in improving the decision-making process, it is applied to guide retention and effective operations [3]. The study points out that the demographic characteristics of the age, the time when the job was taken and the job satisfaction are among the most significant influencing the employee turnover [4].

As it is common knowledge, the attrition can be predicted with the use of a variety of ML algorithms. As indicators, one of the researches [5] compared the performance of the ensemble process, and among the others, the XGBoost and random forest provided the following results: the accuracy that could predict an accuracy rate of 89-percent of the data of 20 features using the XGBoost. The deep learning techniques have also achieved as much as 92 per cent but are usually difficult in terms of the contention of the models [6]. By contrast, simple and logistic regression models occupy a relatively desirable underdog position since it is easy to operate, and it is competing with the other two in the accuracy of 85% [7].

Another characteristic of the models that is significant to reinforce is preprocessing and preselection. Selection of component of influential features is applicable in order to optimize the precision of forecast and reduce the complexity of calculations as it was proven by [8]. In the same argument, it was determined in a comparative study that decision tree models were more suitable than Naive Bayes which provided 82.7 percent accuracy of percentage split evaluation and logistic regression which also provided 85 percent accuracy [10] at percentage split evaluation.

Despite these advances, the strategies also have certain loopholes in dealing with the problem of the unequal distribution of classes and its progressively interpretable character among HR practitioners. The modern research endeavors to overcome these shortcomings by applying the XGBoost with a more refined preprocessing, hyperparameter optimization, and estimation of the importance of the features with the aim of developing the right forecasts and practical knowledge of variables influencing the staff turnover.

3. Methodology

This section establishes the research methodology that comprises data set, pre-processing, type of model and the measures of evaluation. The framework is formulated in a manner that is strong and they can be reproducible and to learn what factors lead to employee attrition. The steps of the process were computed in such a way that it would provide maximum accuracy to the model and it would not be overly complicated to understand by the HR practitioners. In addition, the methodology is associated with openness and the research can be replicated and generalized in the future.

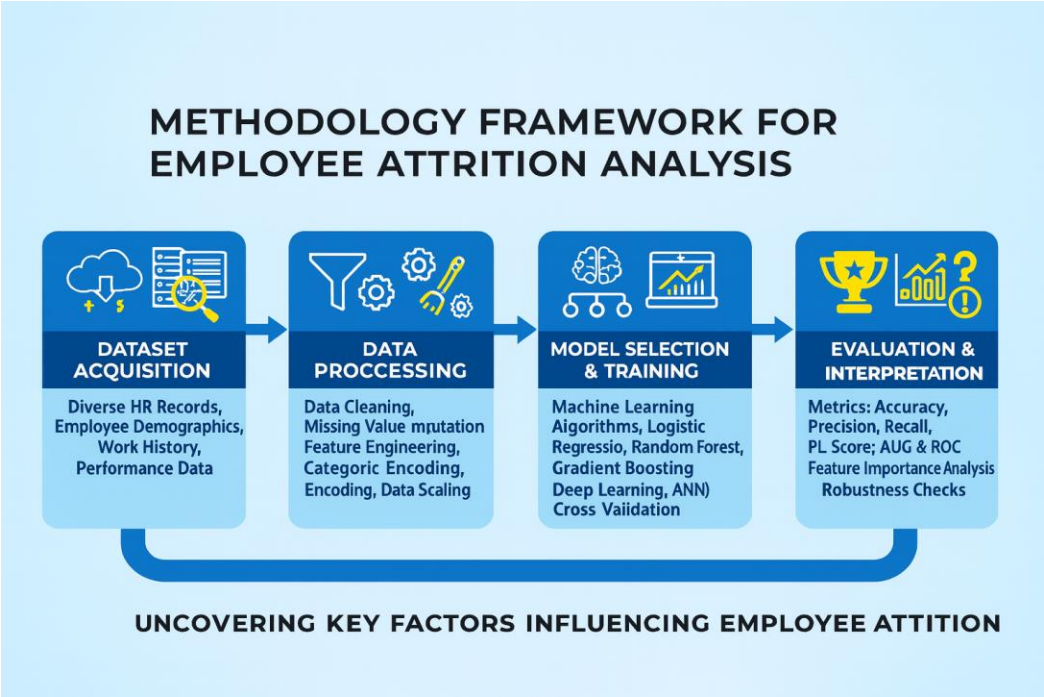


Figure 1: Phases of analysis process

3.1. Dataset

The dataset that was used in this paper is the IBM HR Analytics Employee Attrition and Performance dataset that was downloaded in the Kaggle repository. The data set will contain 1470 data records on workers that are demographic, organizational and performance based. Some of the valuable features include age, sex, department, job position, overtime, monthly earnings and job satisfaction. The target variable is the attrition that is a binary variable (Yes = 1, No = 0) meaning that the employee has quit the organization or not [1].

	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	JobInvolve
t	1470	1470.00	1470	1470.00	1470	1470.00	1470.00	1470	1470.00	1470	14
e	2	NaN	3	NaN	3	NaN	NaN	6	NaN	2	
o	No	NaN	Travel_Rarely	NaN	Research & Development	NaN	NaN	Life Sciences	NaN	Male	
q	1233	NaN	1043	NaN	961	NaN	NaN	606	NaN	882	
n	NaN	36.92	NaN	802.49	NaN	9.19	2.91	NaN	2.72	NaN	
d	NaN	9.14	NaN	403.51	NaN	8.11	1.02	NaN	1.09	NaN	
n	NaN	18.00	NaN	102.00	NaN	1.00	1.00	NaN	1.00	NaN	
6	NaN	30.00	NaN	465.00	NaN	2.00	2.00	NaN	2.00	NaN	
6	NaN	36.00	NaN	802.00	NaN	7.00	3.00	NaN	3.00	NaN	
6	NaN	43.00	NaN	1157.00	NaN	14.00	4.00	NaN	4.00	NaN	
x	NaN	60.00	NaN	1499.00	NaN	29.00	5.00	NaN	4.00	NaN	

Figure 2: Dataset descriptive statistics

3.2. Data Preprocessing

To ensure data quality and reliable model performance, several preprocessing steps were applied:

3.2.1. Handling Missing Values

Missing values in rows were eliminated in order to ensure consistency and eliminate the chances of bias during model training. Even though data imputation processes can be used to enhance the completeness of a dataset, the research in this study chose to omit the rows because of the low percentage of omissions, thus reducing the possibilities of distortion of data [6].

3.2.2. Encoding Categorical Variables

Categorical variables, such as gender, department, job role and overtime, were encoded using one-hot encoding methods to assign them numbers. Such a strategy facilitated the coding of labels whereby label encoding of the target variable Attrition was applied i.e. the yes answer was coded as 1 and the no answer was encoded as 0 which puts the variable in a binary format that can be utilized in classification processes [2].

3.2.3. Train-Test Split

To test the possibility of the generalization, the data was separated into training and testing (20 and 80) sets. The stratified sampling was used to even out the classes of the non-attrition and attrition cases. It was also seeded randomly to enhance comparability and reproducibility or experiment [5, 8].

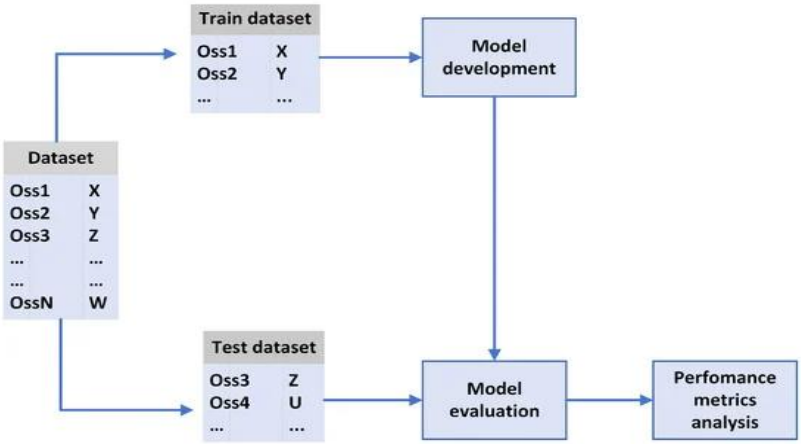


Figure 3: Data preprocessing

3.2.4. Model Selection

The use of Extreme Gradient Boosting (XGBoost) classifier as the primary model is due to its efficiency, scaling, and imbalance data features [2,5]. The XGBoost has already been applied in the area of employee attrition, and the research has demonstrated that it is a great predictor and that it can succeed in a broad spectrum of organizational data [3,6].The XGBoost has been compared with the Logistic Regression, Decision Trees and Random Forests, and Support Vector Machines (SVMs), which were also considered in the studies [3,5,7]. The second comparative framework ensures that the predictive capability of XGBoost is validated against the other simpler and established classifiers and enhances the legitimacy of the model choice [1,3,5].

3.2.5. Model Training Hyperparameter Tuning

The initial parameters used to train XGBoost classifier were default. Significant hyperparameters including the learning rate, max depth, the number of estimators and subsample were optimized using grid search and cross validation in order to have more predictive accuracy and reduced overfitting [2,5]. In the training of a model, feature importance scores were also acquired in order to find out the most important predictors of employee departure. It is also interesting to note that the characteristics such as OverTime, JobSatisfaction and MonthlyIncome were significant and yielded useful data on the organizational variables that had the strongest relationships with the employee turnover [6,11,12].

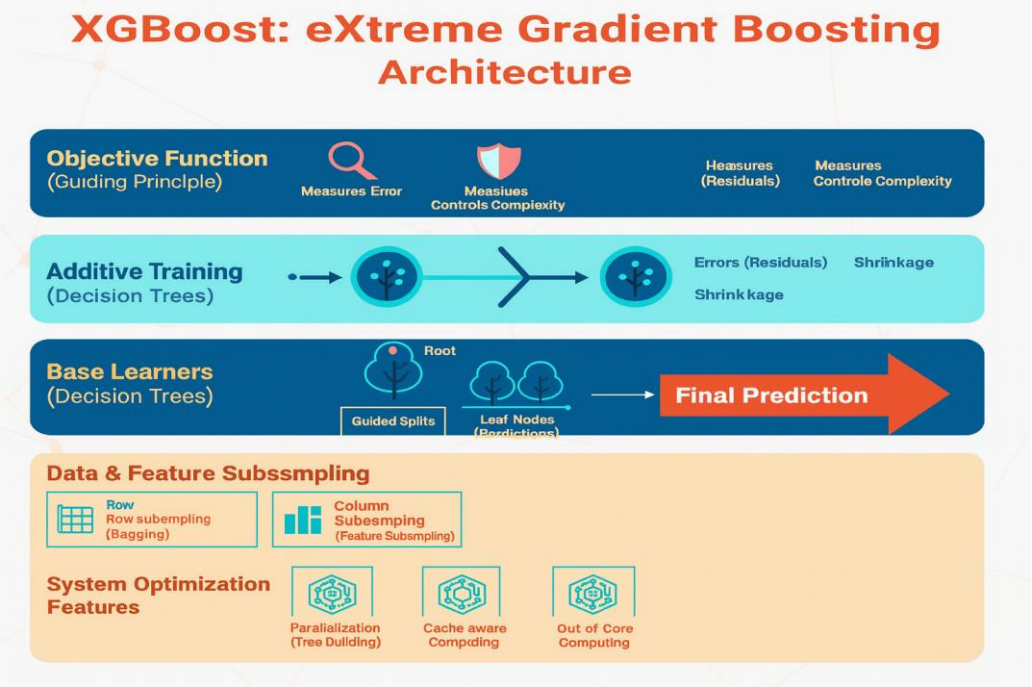


Figure 4: XGBoost Architecture

3.2.6. Feature Importance Analysis

Parameters learned by XGBoost classifier include the following: learning rate, maximum depth, the number of estimators and sub sample available [2,5] with default parameters and trained on grid search and cross-validation. The Python packages such as Scikit-learn, XGBoost API, and Pandas made it possible to ensure the reproducibility of the results because the feature importance was checked on the basis of the OverTime_Yes, JobSatisfaction, and MonthlyIncome as important predictors of attrition. The greater the working hours the greater the risk of turnover, the greater the job satisfaction the lesser the risk of turnover, the greater the competitive income the greater the retention [6,11,12]. These are the ones that are consistent with the previous studies and can be implemented in HR activities like managing workloads, job satisfaction programs, and fair remunerations.

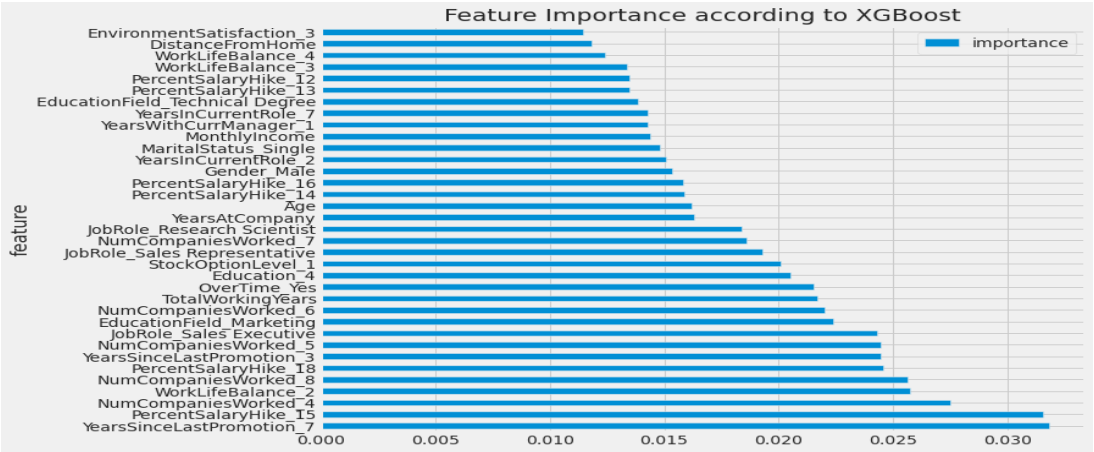


Figure 5: Feature importance score

3.2.7. Evaluation Metrics

In order to check the stability of the XGBoost model and practicability of the model, the following standard measures of classification were used:

- **Accuracy:** The mean percentage of appropriately classified instances that gives a relative estimate of the model performance.
- **Precision:** This metric is evaluated as a measurement of how the model predicts positive instances and also measures the false positives.
- **Recall:** This is an evaluation used to determine how the model predicts all the true positive instances.
- **F1-Score**: Gives a more specific picture of precision and remembrance, and especially with unbalanced data.
- **Confusion matrix:** A row and a column in a confusion matrix will depict true positives, true negatives, false positives and false negatives.

4. Results and Discussion

The high performance of the model was exhibited by the classifier having the training accuracy of 99.87 percent and the test accuracy of 87.76 percent, which was trained using XGBoost. The large value of the training accuracy indicates that the model is doing well in learning patterns with the help of the training data, but the rather small value of the testing accuracy indicates the difficulty in transferring the results to the data that is not observed. The measures of evaluation are condensed in Table 1.

Metric	Training Set (%)	Testing Set (%)
Accuracy	99.87	87.76
Precision (Class 0)	99.94	90.00
Recall (Class 0)	99.91	97.00
F1-Score (Class 0)	99.92	93.00
Precision (Class 1)	99.70	59.00
Recall (Class 1)	99.80	26.00
F1-Score (Class 1)	99.75	36.00

Table 1: Model Evaluation Metrics

5. Conclusion

As analyzed in this paper machine learning and XGBoost, in particular, will prove handy when predicting employee attrition. The model provided a training and testing ratio of 99.87 and 87.76 respectively and the important predictors in the model were OverTime_Yes, JobSatisfaction and MonthlyIncome among others.

Even though the overall performance is high, however, there is an issue with anticipating the minority class (those who leave) and it is founded on the necessity to regulate the imbalance in classes in future employment. Finally, the study is noteworthy because it demonstrates the possibility of changing the employee retention process with the help of machine learning to present the primary causes of turnover and the opportunity to make active decisions of the human resources, as well as to address ethical concerns, including data privacy, and equity.More research is needed to understand how the model will help in other fields, use modern tools, including deep learning or ensemble models, and rationalize ethical considerations, including data privacy, and fairness.

Funding Statement: Author has not received any funds from external source.

Conflicts of Interest: Author has no conflicts of interest.

Data Availability: The IBM HR Analytics dataset used in this study is available publicly.

References

[1] Analytics Vidhya. (2021). Employee Attrition Prediction - A Comprehensive Guide. Retrieved from https://www.analyticsvidhya.com/blog/2021/11/employee-attrition-prediction-a-comprehensive-guide/

[2] Fallucchi, Francesca, Marco Coladangelo, Romeo Giuliano, and Ernesto William De Luca. "Predicting employee attrition using machine learning techniques." *Computers* 9, no. 4 (2020): 86.

[3] Raza, Ali, Kashif Munir, Mubarak Almutairi, Faizan Younas, and Mian Muhammad Sadiq Fareed. "Predicting employee attrition using machine learning approaches." *Applied Sciences* 12, no. 13 (2022): 6424.

[4] Patil, Harsh, and Prabha Kadam. "Machine Learning Applications in Human Resource Management: Predicting Employee Turnover and Performance." *The Voice of Creative Research* 7, no. 2 (2025): 295-301.

[5] Alshiddy, Muneera Saad, and Bader Nasser Aljaber. "Employee attrition prediction using nested ensemble learning techniques." *International Journal of Advanced Computer Science and Applications* 14, no. 7 (2023).

[6] Sari, Sindi Fatika, and Kemas Muslim Lhaksmana. "Employee attrition prediction using feature selection with information gain and random forest classification." *Journal of Computer System and Informatics (JoSYC)* 3, no. 4 (2022): 410-419.

[7] Ponnuru, S. R., G. K. Merugumala, Srinivasulu Padigala, Ramya Vanga, and Bhaskar Kantapalli. "Employee attrition prediction using logistic regression." *International Journal for Research in Applied Science and Engineering Technology* 8, no. 5 (2020): 2871-2875.

[8] Ali, Zeravan Arif, Ziyad H. Abduljabbar, Hanan A. Tahir, Amira Bibo Sallow, and Saman M. Almufti. "eXtreme gradient boosting algorithm with machine learning: A review." *Academic Journal of Nawroz University* 12, no. 2 (2023): 320-334.

[9] Pristyanto, Yoga, Zulfikar Mukarabiman, and Anggit Ferdita Nugraha. "Extreme gradient boosting algorithm to improve machine learning model performance on multiclass imbalanced dataset." *JOIV: International Journal on Informatics Visualization* 7, no. 3 (2023): 710-715.

[10] Usha, P. M., and N. V. Balaji. "A comparative study on machine learning algorithms for employee attrition prediction." In *IOP Conference Series: Materials Science and Engineering*, vol. 1085, no. 1, p. 012029. IOP Publishing, 2021.

[11] Al-Suraihi, Walid Abdullah, Siti Aida Samikon, Al-Hussain Abdullah Al-Suraihi, and Ishaq Ibrahim. "Employee turnover: Causes, importance and retention strategies." *European Journal of Business and Management Research* 6, no. 3 (2021): 1-10.

[12] Tnay, Evelyn, Abg Ekhsan Abg Othman, Heng Chin Siong, and Sheilla Lim Omar Lim. "The influences of job satisfaction and organizational commitment on turnover intention." *Procedia-Social and Behavioral Sciences* 97, no. 201-208 (2013): 3-8.