WEATHER PREDICTION BY USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

P. MUTHUPANDEESWARI (923819106031)
A. INDUMATHI (923819106019)
K. KEERTHANA (923819106023)
R. SHARMILA DEVI (923819106047)

in partial fulfillment for the award of the degree
of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

MANGAYARKARASI COLLEGE OF ENGINEERING, MADURAI

ANNA UNIVERSITY: CHENNAI 600 025

April/May 2023

ANNA UNIVERSITY: CHENNAI 600 025

April 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled "**WEATHER PREDICTION BY USING MACHINE LEARNING**" is the bonafide work of **P.MUTHUPANDEESWARI(923819106031),A.INDUMATHI(923819106019),K.KEERTHANA(923819106023),R.SHARMILADEVI(923819106047)** who Carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Mr.T.VIGNESH,M.E.,MISTE	Mr.I.JESUDASS, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
ASSOCIATE PROFESSOR	ASSISTANT PROFESSOR
Department of ECE,	Department of ECE,
Mangayarkarasi College of Engineering,	Mangayarkarasi College of Engineering,
Paravai, Madurai – 625 402.	Paravai, Madurai – 625 402.

Submitted for the Project Viva – Voce examination held on_________________

…………………… ……………………

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We would like to express our heartiest thanks to God for giving success in all our work.

We take this opportunity to express our sincere thanks to our most respected Secretary **Dr.P.ASHOK KUMAR,M.A.,M.Ed.,BGL.,** who is the guiding light for all the activities in our College.

We would like to express our special thanks of gratitude to our beloved Director **Er.A.SHAKTI PRANESH,B.E.,MBA.,** as well as our principal **Dr.J.KARTHIKEYAN,M.E.,Ph.D.,MBA.,MIE.,MISTE.,C.Eng.,** who gave us the golden opportunity to do this wonderful project.

We would like to extent our heartfelt gratitude to our respected Vice Principal **Mr.T.VIGNESH,M.E.,MISTE.,** as well as our Academic Dean **Dr.C.CALLINS CHRISTYANA,BE.,M.Tech.,Ph.D.** who kind-heartedly permitted us to go ahead with our project.

We express our sincere thanks to project coordinator and Head of the Department **Mr.T.VIGNESH,M.E.,MISTE.,** for his cooperation, guidance and suggestions at every stage of our project work.

We acknowledge our sincere thanks to our inspired guide, **Mr.I.JESUDASS,M.E.**, Assistant Professor, Department of Electronics and Communication Engineering, for his valuable suggestions, inspiration and support to undertake this project work.

Finally, yet importantly, We would like to express our heartfelt thanks to our beloved parents for their blessings, all other faculty of ECE department, our friends and classmates for their help and wishes for the successful completion of this project.

ABSTRACT

Weather prediction has always been a complex problem due to the large number of variables involved, such as temperature, humidity, wind speed, and precipitation, among others. In recent years, machine learning techniques have been increasingly used to improve weather forecasting accuracy. This paper reviews the current state of the art in machine learning-based weather prediction, including the use of deep learning, neural networks, and ensemble techniques. We also discuss the challenges faced in weather prediction and the potential for future research. Finally, we present a case study on the application of machine learning to predict rainfall in a particular region and evaluate the results using metrics such as mean absolute error and root mean squared error. Our results show that machine learning techniques can significantly improve the accuracy of weather prediction, and have the potential to revolutionize the field of meteorology.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 General

Weather forecasting means predicting the weather and telling how the weather changes with change in time. Change in weather occurs due to movement or transfer of energy. Many meteorological patterns and features like anticyclones, depressions, thunderstorms, hurricanes and tornadoes occur due to the physical transfer of heat and moisture by convective processes. Clouds are formed by evaporation of water vapor. As the water cycle keeps on evolving the water content in the clouds increases which in turn leads to precipitation. This is how the convective process happens and also the change in weather. Many factors like temperature, rainfall, pressure, humidity, sunshine, wind and cloudiness are considered for predicting the weather. It is also possible to identify the different types of clouds associated with different patterns of weather. These patterns of weather help in predicting the weather forecast.

In the past, people used barometric pressure, current weather conditions, sky condition to predict whereas now there are many computer based models that consider the atmospheric factors to predict the weather. These methods are not accurate and the reason is due to the chaotic nature of the atmosphere as it keeps on changing. Even predicting weather for a longer period of time will not be accurate that is why most of the current forecasting models predict weather only for a couple of days not more than . The accuracy gets reduced with increase in time.

Weather forecasting isn't a purely mechanical linear process, that standard practices and procedures will be directly applied. Forecaster's job is predicated on theoretical background and lab work which needs several years of study but mainly day-to-day practice inside a weather forecasting service having a particular technical environment. The work of the forecasters has evolved significantly over the years to require advantage of both scientific and technological improvements. The skill of numerical models has improved such a lot that some centers are automating routine forecasts to permit forecasters to specialize in high impact weather or areas where they can add significant value. So it's dangerous to see a regular thanks to achieve weather forecasts.

1.2 Brief explanation of the purpose and scope of the project

1.2.1 Purpose

Every Human subject to adjust themselves with respect to weather condition for their dressing habits to strategic organizational planning activities, since the adverse weather conditions may cause a considerable damage to lives and properties. We need to be on alert for these adverse weather conditions by taking some precautions and using prediction mechanisms to detect them and provide early warning of hazardous weather phenomena. Weather prediction is an indispensable requirement for all of us. Weather is important for most aspects of human life. Predicting weather is very useful. Humans have attempted to make predictions about the weather, many early religions used gods to explain the weather. Only relatively recently have humans developed reasonably accurate weather predictions. We decided to collect weather data and measured the accuracy of predictions made using linear regression. The Weather prediction model designed by us would be of great use to the farmers and for normal being as well. This model basically uses historical weather data to predict the weather on a specific day of and year in the future. Initially the aim is to teach the model with large historical data set and then use it for weather prediction.

1.2.2 Scope

There is a general and increasing interest on weather information, since every day we habitually give an ear to weather forecast news for local and large-scale long-term or short-term weather predictions. Leading weather research institutions and companies have been developing weather prediction systems capable of detecting, predicting and forecasting weather phenomena and hazards by utilizing state-of-the science technologies. Thus weather prediction utilization fields and prediction accuracy increases monotonically by the time.

1.3 Importance of weather prediction and its impact on various sectors

Weather prediction is the process of using scientific techniques and data analysis to forecast the weather conditions for a specific time and location. It plays a crucial role in various sectors and has a significant impact on human activities and natural systems.

Here are some of the sectors that rely on accurate weather prediction:

Agriculture: Farmers need accurate weather information to plan their planting and harvesting schedules, optimize crop yields, and protect their crops from severe weather conditions.

Transportation: Weather conditions can affect transportation routes and schedules, leading to delays, cancellations, and safety concerns. Accurate weather prediction allows transportation companies to plan their routes and schedules accordingly and ensure the safety of passengers and cargo.

Energy: Weather conditions such as wind, solar radiation, and temperature can affect energy production and consumption. Accurate weather prediction enables energy companies to optimize their production and distribution systems and ensure the reliability and stability of the energy grid.

Construction: Weather conditions can affect construction projects, leading to delays, cost overruns, and safety concerns. Accurate weather prediction enables construction companies to plan their schedules and operations accordingly and ensure the safety of workers and equipment.

Emergency management: Weather conditions can pose a significant threat to public safety, leading to disasters such as floods, hurricanes, and wildfires. Accurate weather prediction allows emergency management agencies to plan their response and evacuation strategies and mitigate the impact of natural disasters.

Overall, accurate weather prediction is essential for various sectors and has a significant impact on human activities and natural systems. It enables decision-makers to plan their operations, reduce risks, and ensure the safety and well-being of individuals and communities.

1.1 **Machine Learning**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. Machine learning is closely related to computational statistics, which focuses on making predictions using computers.

The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is related field of study, focusing the exploratory data analysis using unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning techniques: Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. For early tasks that humans assigned to computers, it was possible to create algorithms telling the machine how to execute all needed steps to solve the problem in hand. So on the computer's part, no learning was needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than have human programmers specify every needed step. Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. These were:

1) Supervised Learning
2) Unsupervised Learning
3) Semi Supervised Learning
4) Reinforcement Learning

1.4.1. Supervised Learning :

Supervised learning algorithms are trained using labeled examples, such as an input where the desired output is known. For example, a piece of equipment could have data points labeled either “F” (failed) or “R” (runs). The learning algorithm receives a set of inputs along with the corresponding correct outputs, and the algorithm learns by comparing its actual output with correct outputs to find errors. It then modifies the model accordingly. Through methods like classification, regression, prediction and gradient boosting, supervised learning uses patterns to predict the values of the label on additional unlabeled data. Supervised learning is commonly used in applications where historical data predicts likely future events. For example, it can anticipate when credit card transactions are likely to be fraudulent or which insurance customer is likely to file a claim Thinking of supervised learning with the concept of function approximation, where basically we train an algorithm and in the end of the process we pick the function that best describes the input data, the one that for a given X makes the best estimation of y (X -> y). Most of the time we are not able to figure out the true function that always make the correct predictions and other reason is that the algorithm rely upon an assumption made by humans about how the computer should learn and this assumptions introduce a bias. The human experts acts as the teacher where we feed the computer with training data containing the input/predictors and we show it the correct answers (output) and from the data the computer should be able to learn the patterns. Supervised learning algorithms try to model relationships and dependencies between the target prediction output and the input features such that we can predict the output values for new data based on those relationships which it learned from the previous data sets. Some common algorithms are:

1) Linear Regression
2) Logistic Regression
3) Decision Tree
4) Random Forest
5) KNN
6) SVM
7) Naïve Bayes

1.4.1.1 Linear Regression

To understand the working functionality of this algorithm, imagine how you would arrange random logs of wood in increasing order of their weight. There is a catch; however – you cannot weigh each log. You have to guess its weight just by looking at the height and girth of the log (visual analysis) and arrange them using a combination of these visible parameters. This is what linear regression is like. In this process, a relationship is established between independent and dependent variables by fitting them to a line. This line is known as the regression line and represented by a linear equation Y= a *X + b. Where Y is the Dependent Variable a is the Slope X is the Independent Variable b is the Intercept The coefficients a & b are derived by minimizing the sum of the squared difference of distance between data points and the regression line.

Real life applications of Linear Regression

1)Risk Management in financial services or insurance domain
2) Predictive Analytics
3) Econometric
4) Epidemiology
5) Weather data analysis
6) Customer survey results analysis

1.4.1.2 Logistic Regression

Logistic Regression is used to estimate discrete values (usually binary values like 0/1) from a set of independent variables. It helps predict the probability of an event by fitting data to a logistic function. It is also called logistic regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected). It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. The sigmoid function, also called the logistic function, gives an 'S' shaped curve that can take any real-valued number and map it into a value between 0 and 1. 0 - (x) = 1/ 1+epow(-x) If the curve goes to positive infinity, y predicted will become 1. If the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it like 0 or NO. If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that patient will suffer from cancer.

Real life applications of Logistic Regression:

1) Cancer Detection

2) Trauma and Injury Severity Score

3) Image Segmentation and Categorization

4) Geographic Image Processing

5) Handwriting recognition

6) Prediction whether a person is depressed based on bag of words from the corpus.

1.4.1.3 DECISION TREE

A decision tree is a decision support tool that uses a tree-like model of decision-making process and the possible consequences. It covers event outcomes, resource costs, and utility of decisions. Decision Trees resemble an algorithm or a flowchart that contains only conditional control statements. A decision tree is drawn upside down with the root node at top. Each decision tree has 3 key parts: a root node, leaf nodes, branches. In a decision tree, each internal node represents a test or an event. Say, a heads or a tail in a coin flip. Each branch represents the outcome of the test and each leaf node represents a class label — a decision taken after computing all attributes. The paths from root to leaf nodes represent the classification rules. Decision trees can be a powerful machine learning algorithm for classification and regression. Classification tree works on the target to classify if it was a heads or a tail. Regression trees are represented in a similar manner, but they predict continuous values like house prices in a neighborhood. The best part about decision trees:

1) Handle both numerical and categoric data

2) Handle multi-output problems

3) Decision trees require relatively less effort in data preparation

4) Nonlinear relationships between parameters do not affect tree performance

Real life applications of Decision Trees

1) Selecting a flight to travel

2) Predicting high occupancy dates for hotels

3) Number of drug stores nearby was particularly effective for a client X

4) Cancer vs non-cancerous cell classification where cancerous cells are rare say (1% 5) Suggest a customer what car to buy.

1.4.1.4 RANDOM FOREST

Random Forests in machine learning is an ensemble learning technique about classification, regression and other operations that depend on a multitude of decision trees at the training time. They are fast, flexible, represent a robust approach to mining high-dimensional data and are an extension of classification and regression decision trees we talked about above. A random forest should have a number of trees between 64–128 trees. Ensemble learning, in general, can be defined as a model that makes predictions by combining individual models. The ensemble model tends to be more flexible with less bias and less variance. Ensemble Learning has two popular methods as:

1) Bagging : Each individual tree to randomly sample from the dataset and trained by s random subset of data, resulting in different trees.

2) Boosting: Each individual tree /model learns from mistakes made by the previous model and improves Random forest run times are quite fast. They are pretty efficient in dealing with missing and incorrect data. On the negatives, they cannot predict beyond the defined range in the training data, and that they may over-fit data sets that are particularly noisy.

Real life applications of Random Forests

1) Fraud detection for bank accounts, credit card

2) Detect and predict the drug sensitivity of a medicine

3) Identify a patient’s disease by analyzing their medical records

4) Predict estimated loss or profit while purchasing a particular stock.

1.4.1.5 KNN

K- nearest neighbor (KNN) is a simple supervised machine learning algorithm that can be used to solve both classification and regression problems. KNN stores available inputs and classifies new inputs based on a similar measure i.e. the distance function. KNN has found its major application in statistical estimation and pattern recognition. KNN works by finding the distances between a query and all inputs in the data. Next, it selects a specified number of inputs, say K, closest to the query. And then it votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

The KNN Algorithm:

1) Load the data

2) Initialize k to a chosen number of neighbors in the data

3) For each example in the data, calculate the distance between the query example and the current input from the data

4) Add that distance to the index of input to make an ordered collection

5) Sort the ordered collection of distances and indices in ascending order grouped by distances

6) Pick the first K entries from the sorted collection

7) Get the labels of the selected K entries

8) If regression, return the mean of the K labels; If classification, return the mode of the K labels

Real world applications of KNN

1) Fingerprint detection

2) Forecasting stock market

3) Currency exchange rate 23

4) Bank bankruptcies

5) Credit rating

6) Loan management

7) Money laundering analyses

8) Estimate the amount of glucose in the blood of a diabetic person from the IR absorption spectrum of that person’s blood.

9) Identify the risk factors for a cancer based on clinical & demographic variables.

1.4.1.6 SVM

SVM stands for Support Vector Machines. Machine learning largely involves predicting and classifying data. To do so, have a set of machine learning algorithms to implement depending on the dataset. One of these ML algorithms is SVM. The idea being simple: create a line or a hyperplane which separates the data into multiple classes. Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. SVM transforms your data base on it, finds an optimal boundary between the possible outputs. Support Vector Machine performs classification by finding the hyperplane that maximizes the margin between the two classes.The vectors that define the hyperplane are called the support vectors. The SVM Algorithm: 1) Define an optimal hyperplane with a maximized margin 2) Map data to a high dimensional space where it is easier to classify with linear decision surfaces 3) Reformulate problem so that data is mapped implicitly into this space.

Real Life Applications of SVM

1) Face detection — classify between face and non-face areas on images

2) Text and hypertext categorization

3) Classification of images

4) Bioinformatics — protein, genes, biological or cancer classification.

5) Handwriting recognition

6) Drug Discovery for Therapy. (In recent times, SVM has played a very important role in cancer detection and its therapy with its application in classification).

1.4.1.7 NAÏVE BAYES

Naive Bayes is super effective, commonly-used machine learning classifier. Naive Bayes is in its own a family of algorithms including algorithms for both supervised and unsupervised learning. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. P(A/B) = [P(B/A)*P(A)] / P(B) Naive Bayes (NB) is naive because it makes the assumption that attributes of a measurement are independent of each other. We can simply take one attribute as independent quantity and determine proportion of previous measurements that belong to that class having the same value for this attribute only. Naive Bayes is used primarily to predict the probability of different classes based on multiple attributes. It is mostly used in text classification while mining the data. If you look at the applications of Naive Bayes , 25 the projects you always wanted to do can be best done by this family of algorithms.

Real world applications of Naive Bayes

1) Classify a news article about technology, politics, or sports
2) Sentiment analysis on social media
3) Facial recognition softwares
4) Recommendation Systems as in Netflix, Amazon
5) Spam filtering

1.4.2 Unsupervised Learning

Unsupervised learning is used against data that has no historical labels. The system is not told the "right answer." The algorithm must figure out what is being shown. The goal is to explore the data and find some structure within. Unsupervised learning works well on transactional data. For example, it can identify segments of customers with similar attributes who can then be treated similarly in marketing campaigns. Or it can find the main attributes that separate customer segments from each other. Popular techniques include self-organizing maps, nearest-neighbor mapping, k-means clustering and singular value decomposition. These algorithms are also used to segment text topics, recommend items and identify data outliers. Unsupervised learning is that algorithm where you only have to insert/put the input data (X) and no corresponding output variables are to be put.

The major goal for the unsupervised learning is to help model the underlying structure or maybe in the distribution of the data in order to help the learners learn more about the data. These are termed as unsupervised learning because unlike supervised learning which is shown above there are no correct answers and there is no teacher to this. Algorithms are left to their own devices to help discover and present the interesting structure that is present in the data. Unsupervised learning problems can even be grouped ahead into clustering and association problems. 1) Clustering: A clustering is that problem which indicates what you want to discover and this helps in the inherent groupings of the data, such as grouping the customers based on their purchasing behavior. 2) Association: An association rule is termed to be the learning problem. This is where you would be discovering the exact rules that will describe the large portions of your data. Example: People who buy X are also the one who tends to buy Y. Some common algorithms are:

1) K-means for clustering problems
2) Apriori algorithm for association rule learning problems
3) Principal Component Analysis
4) Singular Value Decomposition
5) Independent Component Analysis

1.4.2.1 K-means CLUSTERING

K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid. K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either the centroids have stabilized or a defined number of iterations have been achieved.

The K-means clustering algorithm:

1) Specify the number of clusters K.

2) Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement

3) Keep iterating until the centroids are stabilized

4) Compute the sum of the squared distance between data points and all centroids

5) Assign each data point to the closest cluster (centroid)

6) Compute the centroids for the clusters by taking the average of the data points that belong to each cluster.

Real World applications of K-means Clustering

1) Identifying fake news

2) Spam detection and filtering

3) Classify books or movies by genre

4) Popular transport routes while town planning

1.4.2.2 APRIORI ALGORITHM FOR ASSOCIATION RULE LEARING PROBLEMS

Apriori is considered an algorithm for frequent itemset mining and association rule learning over transactional databases. It proceeds just by identifying the frequent individual items in the database and then extending them to larger and larger item sets. The observation is, for as long as those item sets appear sufficiently often in the database. The frequent item sets that were determined by Apriori can be later used to determine about the association rules which highlights all the general trends that are being used in the database: this has got applications that fall in the domains such as the market basket analysis.

1.4.2.3 PRINCIPAL COMPONENT ANALYSIS

The main idea which falls behind the principal component analysis (PCA) is to help in reducing the dimensionality of the dataset which consists of many variables, that are always correlated with each other, either in a heavy or light manner, while retaining the variation which is present in the dataset, up to its maximum extent. The same thing is repeated and done by transforming and bringing the variables to a whole new set of variables, which are called the principal components (or simply, the PCs) and are even termed to be orthogonal, ordered in such a way that the retention of variation which is present in the original variables can be decreased as we try to move down in the proper order. So, by following this particular way, the 1st principal component retains the most and maximum variation that was earlier present in the original components. The principal components are basically known to be the eigenvectors of a covariance matrix, and hence they are even called the orthogonal. Most importantly, the dataset which is based on what the PCA techniques are to be used and must be scaled. The result also turns out to be sensitive based on the relative scaling.

As a layman, it can be termed as a method of summarizing data. Just imagine having some wine bottles on your dining table. Each wine would be described only by its attributes, that are like colour, age, strength, etc. But eventually, redundancy will arise maybe because many of them would be measured based on the related properties. Principal component analysis might not be the best candidate in the algorithm category, but it definitely is super-useful as a machine learning technique. Principal Component Analysis (PCA) is an unsupervised, statistical technique primarily used for dimensionality reduction by feature extraction in machine learning. When we talk about high-dimensionality, it means that the dataset has a large number of features. and that requires a large amount of memory and computational power.

PCA uses orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables.It is used to explain the variance-co variance structure of a set of variables through linear combinations. It is a also the most widely used tool in exploratory data analysis and predictive modeling. The idea behind PCA is simply to find a low-dimension set of axes that summarize data. Say for example, we have a dataset composed by a set of car properties; size, color, number of seats, number of doors, size of trunk, circularity, compactness, radius… However, many of these features will indicate the same result and therefore can be redundant. We as smart technologists should try to remove these redundancies and describe each car with fewer properties, making the computation simple. This is exactly what PCA aims to do. PCA does not take information of attributes into account. It concerns itself with the variance of each attribute because the presence of high variance would indicate a good split between classes, and that's how we reduce the dimensionality. PCA never just considers some while discards others. It takes the attributes into account statistically.

Real world applications of PCA

1) Optimize power allocation in multiple communication channels

2) Image Processing

3) Movie recommendation system

1.4.2.4 Singular Value Decomposition

In linear algebra, you can call the singular-value decomposition (SVD) as a factorization of maybe real or complex matrix. It is the generalization of the eigen decomposition, that is the origin of a positive semidefinite normal matrix is done somewhere over here. It has many useful applications that are signal processing and are into statistics. The singular-value decomposition can be computed easily by making the use of the following observations:

- The left-singular vectors of M are considered to be a set of orthonormal eigenvectors of MM∗.

- The right-singular vectors of M are actually the set of orthonormal eigenvectors of M∗M.
- The non-zero singular values of M (that are found on the diagonal entries of Σ) are considered to be the square roots of the nonzero eigenvalues of both M∗M and MM∗. Applications that help to employ the SVD include computing of the pseudoinverse, the least squares fitting of data, multivariable control, matrix approximation, and determining the rank, range and null space of a matrix.

1.4.2.5 INDEPENDENT COMPONENT ANALYSIS

Independent component analysis (ICA), it is considered to be a statistical and computational technique. It helps to bring our or in revealing hidden factors that underlie in the sets of random variables, measurements, or signals. ICA helps to define a generative model. This model stands for the observed multivariate data. It is typically recognized in the form of a large database of samples. Well, In the model, the data variables are assumed to be the linear mixtures of few less known or you can call it as unknown latent variables, and even the mixing system is also unknown. Then comes the latent variables. These variables are actually assumed to be the non-gaussian. They are even the mutually independent ones. These could be termed as the independent components belonging in the category of the observed data. These independent components, also termed as the sources or factors, can be found by the ICA. ICA, the term is basically superficially related to the principal component analysis and then to the factor analysis.

ICA is considered and supposedly it is a much more powerful technique. Still, however this would be always capable of finding the underlying factors. It can even be the sources if possible by any chance, if these classic methods fail completely anyhow. The data which is analyzed by the ICA could be originating from various kinds of application fields, this could be including digital images, the document databases, the economic indicators and then the psychometric measurements. In many cases, these measurements are given to be considered as a set of parallel signals or time series; the term blind source separation is then used in this to characterize this problem. Typical examples are actually the mixtures of simultaneous speech signals that have been picked up by several microphones, these are the brain waves that is recorded by multiple sensors and then the interfering radio signals that arriving at a mobile phone, or maybe the parallel time series which is obtained from performing some industrial process.

1.4.2.6 SEMI SUPERVISED LEARNING

In the previous two types, either there are no labels for all the observation in the dataset or labels are present for all the observations. Semi-supervised learning falls in between these two. In many practical situations, the cost to label is quite high, since it requires skilled human 32 experts to do that. So, in the absence of labels in the majority of the observations but present in few, semi-supervised algorithms are the best candidates for the model building. These methods exploit the idea that even though the group memberships of the unlabeled data are unknown, this data carries important information about the group parameters. Semi supervised learning is used for the same applications as supervised learning. But it uses both labeled and unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data (because unlabeled data is less expensive and takes less effort to acquire). This type of learning can be used with methods such as classification, regression and prediction. Semi supervised learning is useful when the cost associated with labeling is too high to allow for a fully labeled training process. Early examples of this include identifying a person's face on a web cam.

1.4.2.7 REINCEFORCEMENT LEARNING

Reinforcement learning is often used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers through trial and error which actions yield the greatest rewards. This type of learning has three primary components: the agent (the learner or decision maker), the environment (everything the agent interacts with) and actions (what the agent can do). The objective is for the agent to choose actions that maximize the expected reward over a given amount of time. The agent will reach the goal much faster by following a good policy. So the goal in reinforcement learning is to learn the best policy. Reinforcement Learning is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal.

There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms. In the problem, an agent is supposed decide the best action to select based on his current state. When this step is repeated, the problem is known as a Markov Decision Process. In order to produce intelligent programs (also called agents), reinforcement learning goes through the following steps:

1) Input state is observed by the agent.

2) Decision making function is used to make agent perform action.

3) After the action is performed, the agent receives reward or reinforcement from the environment.

4) The state-action pair information about the reward is stored.

CHAPTER 2

LITERATURE SURVEY

2.1 Review of existing studies on weather prediction using machine learning

[1] Paper name: Smart weather prediction using machine learning
Author name: Suvendra Kumar Jayasingh, Jibendu Kumar Mantri and Sipali Pradhan
Description: In this paper, the authors propose a machine learning-based approach for weather prediction using a hybrid model that combines three machine learning algorithms: Support Vector Machine (SVM), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN). The authors use historical weather data to train these models and then use them to predict weather conditions for future time periods. The paper also presents a system architecture for smart weather prediction that includes data collection, data pre-processing, feature extraction, model training and evaluation, and prediction. The authors use Python programming language and several libraries such as Pandas, NumPy, Scikit-learn, and Keras to implement their system. The authors evaluate their proposed approach using real-world weather data from India and compare its performance with traditional statistical methods. The results show that their approach achieves higher accuracy and outperforms traditional methods. Overall, this paper presents an interesting approach to weather prediction using a hybrid model of SVM, ANN, and KNN and provides insights into how this technology can be used to improve the accuracy and efficiency of weather forecasting systems.

[2]Paper Name: Machine learning for post processing ensemble streamflow forecast.

Author Name: Shivam Tandon, Abhishek Patel, Pawan Kumar Singh

Description: In this paper, The authors use historical streamflow data to train a machine learning model that learns the relationships between input variables such as precipitation, temperature, and soil moisture and the streamflow output. They then use the trained model to post-process ensemble streamflow forecasts generated by a hydrological model and correct any biases or errors in the forecasts. The paper presents several machine learning models, including Support Vector Regression (SVR), Random Forest (RF), and Artificial Neural Network (ANN), and compares their performance with traditional statistical methods. The results show that the machine learning-based approach outperforms traditional methods and improves the accuracy and reliability of streamflow forecasts. The paper also discusses the potential applications of the proposed approach in water resources management, flood forecasting, and drought monitoring. The authors suggest that their approach can help decision-makers to make informed decisions and reduce the impacts of extreme weather events on human activities and natural systems. Overall, this paper presents an interesting approach to post-processing ensemble streamflow forecasts using machine learning and highlights the potential benefits of this technology in water resources management and related fields.

[3]Paper Name: Prediction and classification of weather using machine learning

Author Name: William Samuel Sanders

Description: In this paper, the author proposes a machine learning-based approach for predicting and classifying weather conditions using weather data such as temperature, humidity, wind speed, and precipitation. The proposed approach uses multiple machine learning algorithms, including Decision Trees, Random Forest, Naive Bayes, and K-Nearest Neighbors (KNN) to predict the weather conditions. The paper also presents a system architecture for the proposed approach, which includes data collection, data pre-processing, feature extraction, model training, and evaluation. The system uses Python programming language and several libraries such as Pandas, NumPy, and Scikit-learn to implement the machine learning models. The author evaluates the performance of the proposed approach using real-world weather data from the National Oceanic and Atmospheric Administration (NOAA) and compares its accuracy with traditional statistical methods. The results show that the machine learning-based approach outperforms traditional methods and achieves high accuracy in predicting and classifying weather conditions. Overall, this paper presents an interesting approach to weather prediction and classification using machine learning and provides insights into how this technology can be used to improve the accuracy and efficiency of weather forecasting systems.

[4]Paper name: Machine learning technique for weather forecasting

Author name: A H M Jakaria, Md Mosharaf Hossain, Mohammad Ashiqur Rahman

Description: There are several machine learning techniques that can be used for weather forecasting. Some of them are:

Artificial Neural Networks (ANNs): ANNs are a type of machine learning model that are well suited for time series forecasting. They are able to learn complex patterns and relationships in data and can be used to predict weather variables such as temperature, humidity, and precipitation. Support Vector Machines (SVMs): SVMs are a powerful machine learning technique that can be used for both regression and classification problems. They have been successfully applied to weather forecasting, particularly in predicting extreme weather events.

Random Forests: Random forests are an ensemble learning method that combines multiple decision trees to make predictions. They can be used for both regression and classification problems and have been shown to be effective for weather forecasting. Gradient Boosting: Gradient boosting is another ensemble learning technique that has been successfully applied to weather forecasting. It works by combining multiple weak models to create a strong model that can make accurate predictions. Overall, the choice of machine learning technique for weather forecasting depends on the specific problem and the available data. It is important to carefully select the appropriate technique and to evaluate its performance on a validation dataset.

[5]Paper Name: A Novel Model for Weather Forecasting Using Deep Learning

Author Name: Andrew Devanas,Tanja Fransen

Description: The use of deep learning for weather forecasting has become increasingly popular in recent years. In their paper "A Novel Model for Weather Forecasting Using Deep Learning," Andrew Devanas and Tanja Fransen propose a new deep learning model for weather forecasting.The proposed model is based on a Long Short-Term Memory (LSTM) network, which is a type of recurrent neural network (RNN) that is particularly well suited for time series forecasting. The model takes as input a set of historical weather data, including variables such as temperature, humidity, and wind speed, and uses the LSTM network to predict future weather conditions.

2.2 Overview of different machine learning algorithms used in weather prediction

Paper Name	Methodology
Smart weather prediction using machine learning	Random forest Decision Tree Support Vector machine KNN Adaboost Xgboost Gradient Boosting Naïve bayes Logistic regression
Machine learning for post processing ensemble streamflow forecast.	Decision tree KNN Random forest.
Prediction and classification of weather using machine learning	Linear Regression Support vector machine.
Machine learning technique for weather forecasting	Linear Regression, SVM Random forest.
A Novel Model for Weather Forecasting Using Deep Learning	Long Short-Term Memory (LSTM) algorithms is used

Table 2.1

2.3 Comparison of accuracy of different algorithm

Paper name	Accuracy
Smart weather prediction using machine learning	Randomforest:79 Decision Tree:71 Support Vector machine:59 KNN:77 Adaboost:71 Xgboost:79 Gradient Boosting:81 Naïve bayes:73 Logistic regression:78
Machine learning for post processing ensemble streamflow forecast.	Decision tree:71 KNN:77 Random forest:79
Prediction and classification of weather using machine learning	Linear Regression:73 Support vector machine:59
Machine learning technique for weather forecasting	Linear Regression:73 SVM:57 Random forest:79
A Novel Model for Weather Forecasting Using Deep Learning	LongShort-Term Memory(LSTM):67

Table 2.2

2.4 Challenges in weather prediction using machine learning

While machine learning has shown great promise for weather prediction, there are still several challenges that need to be addressed. Some of the main challenges in weather prediction using machine learning are:

Data quality: Machine learning models are only as good as the data they are trained on. Inaccurate or incomplete data can lead to inaccurate predictions. Weather data can be noisy and contain missing values, which can make it difficult to train accurate models.

Limited data: Weather data is often limited, particularly for certain regions or time periods. This can make it difficult to train models that are robust and generalizable.

Non-stationary data: Weather data is highly dynamic and non-stationary, meaning that patterns and relationships can change over time. Machine learning models need to be able to adapt to these changes in order to make accurate predictions.

Computational complexity: Weather prediction involves analyzing large amounts of data over long time periods. Machine learning models can be computationally expensive, which can make it challenging to train and deploy them in real-time.

Interpretability: Machine learning models can be difficult to interpret, which can make it challenging to understand why certain predictions are made. In the context of weather prediction, it is important to be able to explain why a particular weather event is predicted to occur.

Uncertainty: Weather prediction is inherently uncertain, and machine learning models need to be able to account for this uncertainty in their predictions. This can be challenging, as uncertainty can arise from a variety of sources, including model error, data quality, and the complexity of the weather system itself.

Addressing these challenges will be critical for the continued development and improvement of machine learning-based weather prediction models.

CHAPTER 3
METHODOLOGY

3.1 Proposed System

Weather prediction is a complex task that requires the analysis of various atmospheric variables, such as temperature, humidity, wind speed, and precipitation, among others. Machine learning algorithms, particularly convolutional neural networks (CNN), have been proposed as a solution to improve the accuracy of weather predictions.

CNNs are a type of deep learning algorithm that can learn hierarchical representations of data by using convolutional layers. In the case of weather prediction, CNNs can be trained on historical weather data to learn the relationship between different atmospheric variables and their impact on weather patterns. Once trained, the CNN can make predictions on future weather conditions based on current or forecasted atmospheric conditions.

One proposed solution for weather prediction using CNN involves using multiple input variables, such as temperature, humidity, wind speed, and precipitation, to predict a single output variable, such as the likelihood of rainfall. The CNN would be trained on historical weather data to learn the complex relationships between these variables and their impact on rainfallThe CNN would then be used to make predictions on future weather conditions based on the current or forecasted atmospheric conditions. By using machine learning algorithms, such as CNNs, weather predictions can be made more accurate, which can be particularly useful for disaster management, agriculture, and transportation, among other fields.

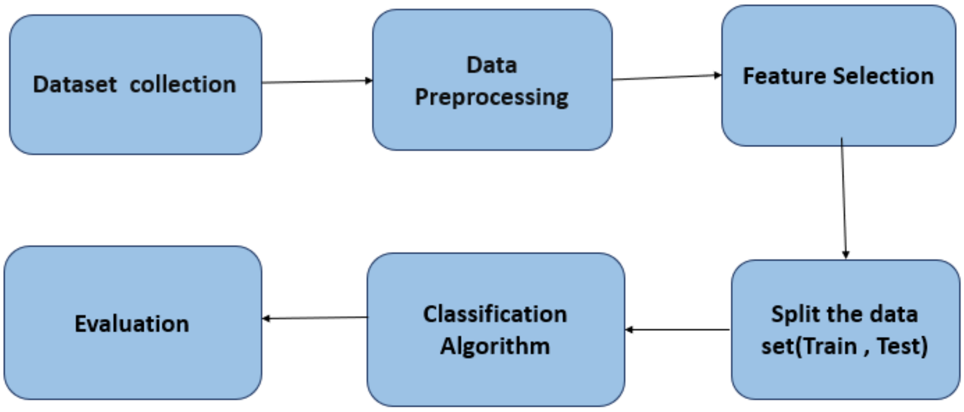


Figure No: 3.1 Proposed Model

3.2 Data collect and Preprocessing

3.2.1 Explanation of the data sources used in the project

Weather prediction models require large amounts of data from various sources to accurately predict weather conditions. Some of the commonly used data sources for weather prediction by using machine learning are:

Atmospheric data: This includes data from weather stations, radars, satellites, and other sensors that collect information on temperature, humidity, wind speed, pressure, and precipitation. This data is usually collected in real-time or near-real-time and can be used to train machine learning models to predict weather conditions.

Historical weather data: Historical weather data is used to train machine learning models to recognize patterns in weather conditions and make predictions. This data is often collected over a long period, such as several decades, and can include information on daily or hourly weather conditions.

Oceanic data: Oceanic data, such as sea surface temperature, can have a significant impact on weather patterns. Machine learning models can use this data to predict changes in weather patterns, such as hurricanes and tropical storms.

Land-use and land-cover data: Changes in land use and land cover, such as deforestation or urbanization, can affect local weather patterns. Machine learning models can use this data to predict how changes in land use will impact weather conditions.

Air quality data: Air quality data can also have an impact on weather patterns, particularly in urban areas. Machine learning models can use this data to predict how changes in air quality will impact weather conditions.

Overall, the accuracy of weather prediction models depends on the quality and quantity of data used to train the machine learning algorithms. Therefore, it is essential to collect and use data from various sources to develop accurate and reliable weather prediction models.

3.2.2 Description of the features used in the prediction model

Weather prediction models typically use a variety of features, or variables, to describe weather conditions. These features can be broadly classified into two categories: atmospheric features and surface features. Atmospheric features include variables that describe the state of the atmosphere, such as temperature, humidity, pressure, wind speed and direction, and precipitation. These variables can be measured at different levels of the atmosphere, such as surface level, upper-level, and mid-level. Surface features include variables that describe the state of the Earth's surface, such as land cover, soil moisture, topography, and sea surface temperature. These variables can have a significant impact on local weather patterns, and machine learning models can use them to predict how changes in surface conditions will impact weather conditions. In addition to these primary features, weather prediction models can also use derived features, which are variables calculated from the primary features. Examples of derived features include dew point, heat index, and wind chill. Machine learning models can use these features to learn the complex relationships between different variables and how they impact weather conditions. The selection and importance of these features can vary depending on the specific weather phenomenon being predicted and the location of the prediction. Overall, the accuracy of weather prediction models depends on the quality and relevance of the features used. Therefore, it is essential to carefully select and preprocess the features to develop accurate and reliable weather prediction models.

3.2.3 Pre-processing techniques used to clean and normalize the data

Pre-processing is a critical step in machine learning that involves cleaning and normalizing the data to prepare it for training machine learning models. Some of the commonly used pre-processing techniques to clean and normalize the data in machine learning are:

Data cleaning: Data cleaning involves identifying and handling missing values, outliers, and errors in the data. Missing values can be imputed using techniques such as mean, median, or regression imputation. Outliers can be detected and removed using statistical methods or domain knowledge. Errors can be corrected by manual inspection or using automated techniques.

Data normalization: Data normalization involves scaling the data to a common range to ensure that all features have equal importance during model training. Common techniques for data normalization include min-max scaling, z-score normalization, and robust scaling.

Feature selection: Feature selection involves identifying the most relevant features for model training while discarding irrelevant or redundant features. This reduces the dimensionality of the data and improves model performance.

Feature transformation: Feature transformation involves transforming the data to improve its distribution or make it more suitable for model training. Common techniques for feature transformation include principal component analysis (PCA), logarithmic transformation, and polynomial transformation.

Data augmentation: Data augmentation involves generating additional data from existing data by applying transformations such as rotation, translation, and scaling.

This technique can help overcome the problem of limited training data and improve model performance.

Overall, the pre-processing techniques used in machine learning depend on the nature of the data and the specific requirements of the model being trained. Proper pre-processing can significantly improve the accuracy and performance of machine learning models.

3.3 Machine Learning Model

3.3.1 Description of the machine learning model used in the project

Convolutional Neural Networks (CNNs) are a type of machine learning model commonly used for image classification and object recognition tasks. CNNs are inspired by the structure and function of the visual cortex in the brain, and they are designed to automatically learn and extract features from raw input data.

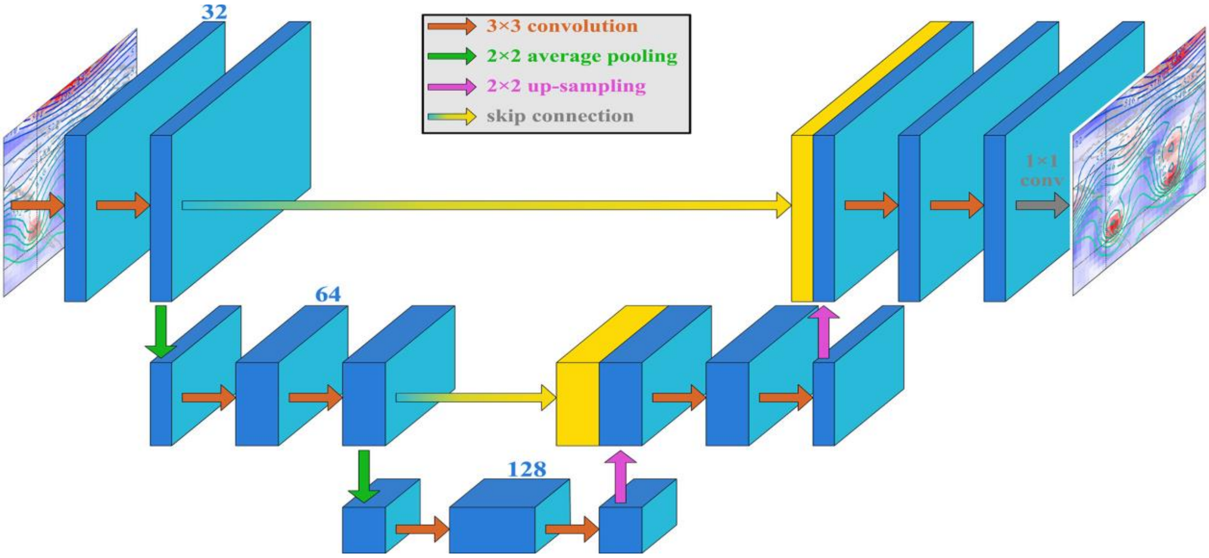
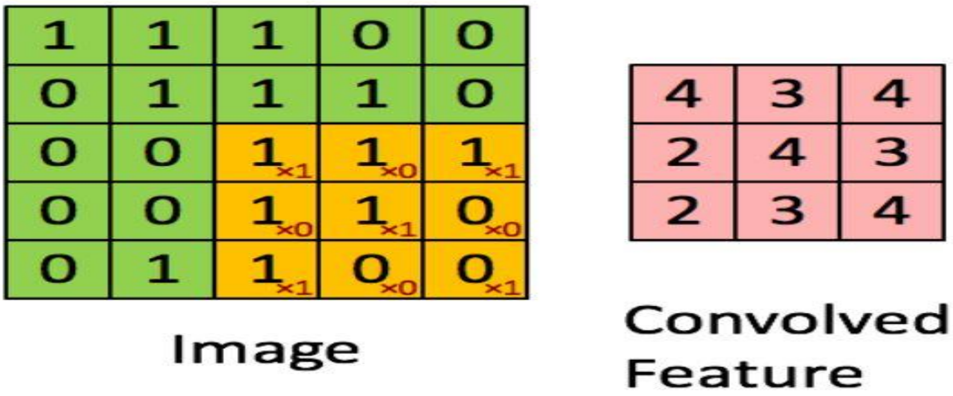


Figure No:3.2 CNN

A typical CNN architecture consists of three main types of layers: convolutional layers, pooling layers, and fully connected layers

Convolutional layers: The Kernel of CNN works on the basis of the following formula. Image Dimensions = n1 x n2 x 1,where n1 = height, n2 = breadth, and 1 = Number of channels such as RGB. So, as an example, the formula will become I D = 5 x 5 x 1. We will explain this using the image given below.



1	1	1	0	0
0	1	1	1	0
0	0	$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$
0	0	$1_{\times 0}$	$1_{\times 1}$	$0_{\times 0}$
0	1	$1_{\times 1}$	$0_{\times 0}$	$0_{\times 1}$

4	3	4
2	4	3
2	3	4

Figure No: 3.3 Convolutional layers

In this image, the green section shows the 5 x 5 x 1 formula. The yellow box evolves from the first box till last, performing the convolutional operation on every 3x3 matrix. This operation is called Kernel (K) and work on the basis of the following binary algorithm.

K =		
1	0	1
0	1	0
1	0	1

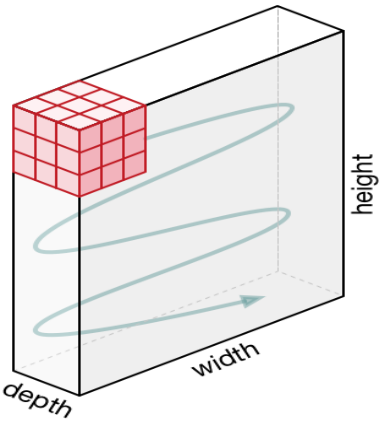


Figure No:3.4 Examples of Convolutional layers

In the above figure, the Kernel moves to the right with a defined value for “Stride.” Along the way, it parses the image objects until it completes the breadth. Then it hops down to the second row on the left and moves just as in the top row till it covers the whole image. The process keeps repeating until every part of the image is parsed.

If there are multiple channels such as found in RGB images, then the kernel contains the same depth as found in the input image. The multiplication of the matrix is implemented based on the number of Ks. The procedure is followed as in stack format, for example, {K1, I1}, {K2, I2}, and so on. The results are generated based on the summation of bias. The result is in the form of a squeezed "1-depth channel" of convoluted feature output.

The goal of this convolution operation is to obtain all the high-level features of the image. The high-level features can include edges of the image too. This layer is not just limited to high-level features; it also performs an operation on low-level features, such as color and gradient orientation. This architecture evolves to a new level and thus includes two more types of layers. The two layers are known as Valid padding and the Same padding.The objective of these layers is to reduce the dimensionality of the image that is found in the original input image and to increase dimensionality or, in some cases, to leave it unchanged, depending on the required output. The same padding is applied to convolute the image to different dimensions of the matrix, while valid padding is applied when there is no need to change the dimension

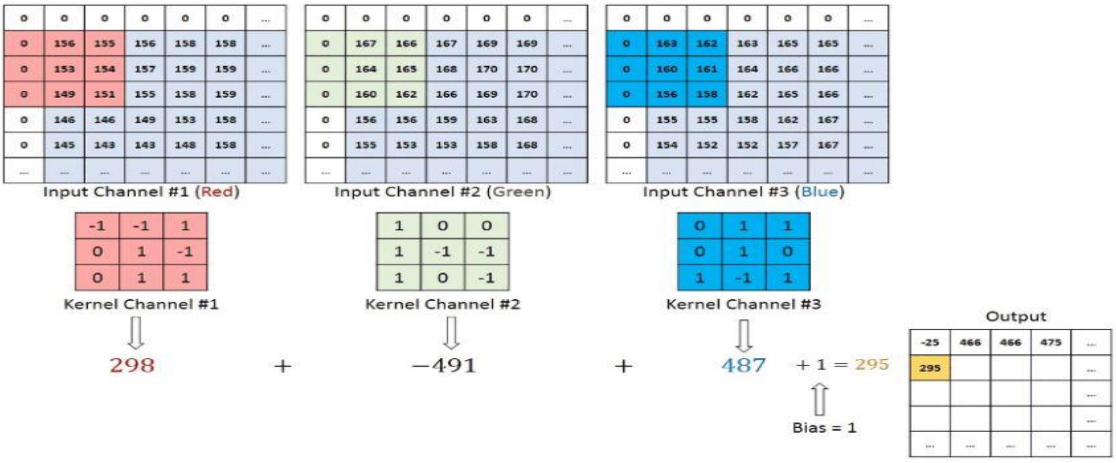


Figure No: 3.5 Examples of convolutional layers

Pooling layers: Pooling layers are used to down sample the feature maps generated by the convolutional layers by aggregating the values in small spatial regions. The most commonly used pooling operation is max pooling, which selects the maximum value in each region. Pooling helps to reduce the dimensionality of the feature maps and makes the model more robust to small spatial variations in the input image.

Fully connected layers: Fully connected layers are used to classify the input image based on the learned features. The output of the last pooling layer is flattened and fed into one or more fully connected layers, which perform a weighted sum of the inputs and produce a probability distribution over the output classes. The final output is obtained by applying a soft max function to the output of the last fully connected layer.

CNNs are trained using a variant of backpropagation called stochastic gradient descent (SGD), which iteratively adjusts the weights of the filters and fully connected layers to minimize the loss function. CNNs are highly effective for image classification tasks due to their ability to learn hierarchical representations of the input data, from low-level features such as edges and corners to high-level features such as shapes and objects.

3.4 Evaluation of the model performance

Evaluation of model performance is a critical step in machine learning that helps assess the accuracy and generalization ability of the trained model. There are several commonly used evaluation metrics for different types of machine learning tasks, such as classification, regression, and clustering.

For classification tasks, common evaluation metrics include:

Accuracy: The percentage of correct predictions made by the model on the test data.

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

Precision: The ratio of true positives to the total number of predicted positives.

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

Recall: The ratio of true positives to the total number of actual positives.

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

F1 score: The harmonic mean of precision and recall.

$$\text{F-measure} = \frac{2*Recall*Precision}{Recall+Precision}$$

For regression tasks, common evaluation metrics include:

Mean Absolute Error (MAE): The average absolute difference between the predicted and actual values.

Mean Squared Error (MSE): The average squared difference between the predicted and actual values.

Root Mean Squared Error (RMSE): The square root of the MSE.

R-squared (R2) score: The proportion of the variance in the dependent variable that is predictable from the independent variable(s).

For clustering tasks, common evaluation metrics include:

Silhouette score: A measure of how similar an object is to its own cluster compared to other clusters.

Calinski-Harabasz Index: A measure of the ratio of between-cluster dispersion to within-cluster dispersion.

It is also important to use cross-validation techniques such as k-fold cross-validation to ensure that the evaluation metrics are not biased due to the particular split of the data into training and test sets. Overall, the choice of evaluation metric depends on the specific machine learning task and the nature of the data. A good evaluation metric should be easy to interpret, sensitive to model performance, and relevant to the problem being solved.

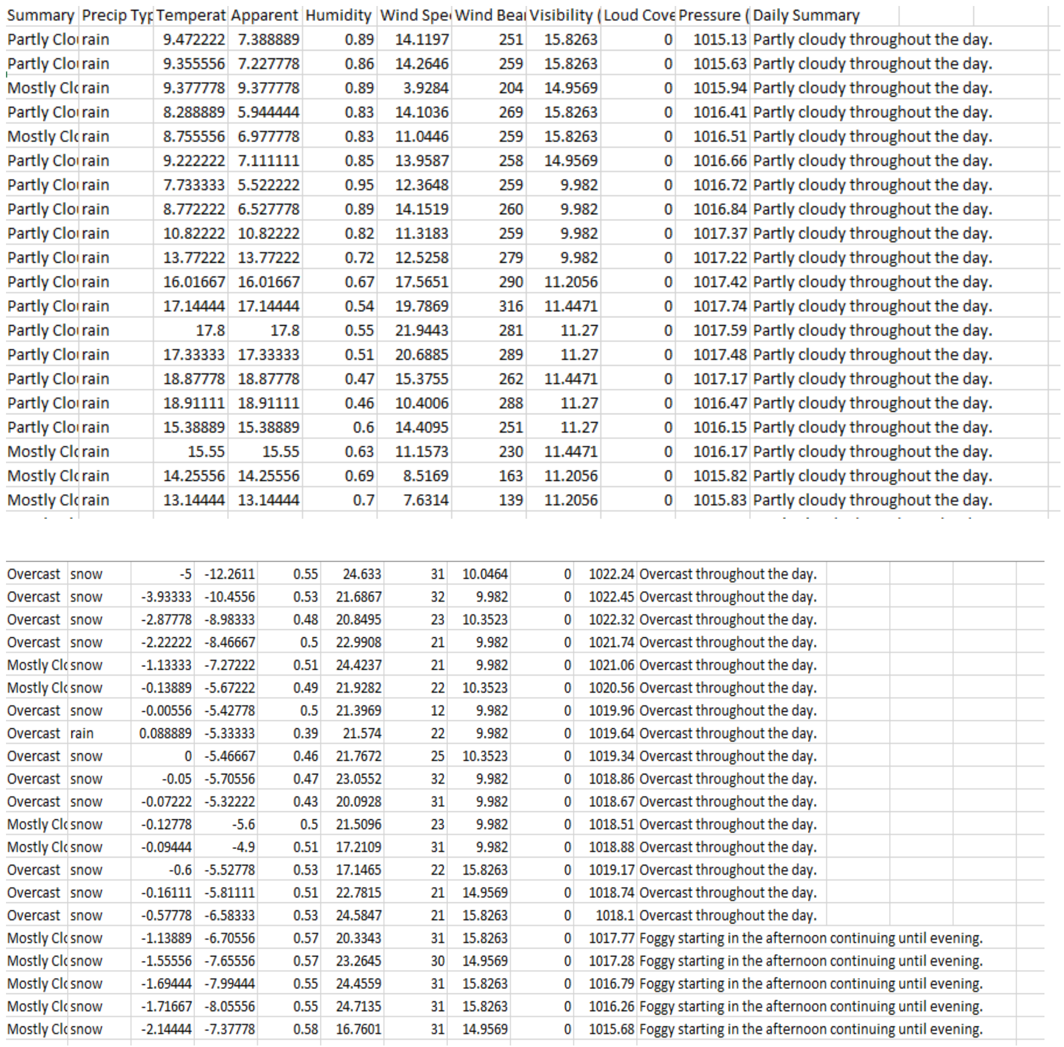
CHAPTER 4

IMPLEMENTATION

4.1 The Dataset

The Data Set we are using has been acquired by a website named https://rp5.ru/

- It provides Present as well as Historical weather data several years back in time depending upon the location of the weather stations.
- The data set provided by the source is kind of reliable and has ample features to work on example mean temp, max temp, humidity, precipitation etc.
- The data set we are using is of a specific area and is categorised date wise.



Summary	Precip Typ	Temperat	Apparent	Humidity	Wind Spe	Wind Bea	Visibility (	Loud Cove	Pressure (	Daily Summary
Partly Clo	rain	9.472222	7.388889	0.89	14.1197	251	15.8263	0	1015.13	Partly cloudy throughout the day.
Partly Clo	rain	9.355556	7.227778	0.86	14.2646	259	15.8263	0	1015.63	Partly cloudy throughout the day.
Mostly Cl	rain	9.377778	9.377778	0.89	3.9284	204	14.9569	0	1015.94	Partly cloudy throughout the day.
Partly Clo	rain	8.288889	5.944444	0.83	14.1036	269	15.8263	0	1016.41	Partly cloudy throughout the day.
Mostly Cl	rain	8.755556	6.977778	0.83	11.0446	259	15.8263	0	1016.51	Partly cloudy throughout the day.
Partly Clo	rain	9.222222	7.111111	0.85	13.9587	258	14.9569	0	1016.66	Partly cloudy throughout the day.
Partly Clo	rain	7.733333	5.522222	0.95	12.3648	259	9.982	0	1016.72	Partly cloudy throughout the day.
Partly Clo	rain	8.772222	6.527778	0.89	14.1519	260	9.982	0	1016.84	Partly cloudy throughout the day.
Partly Clo	rain	10.82222	10.82222	0.82	11.3183	259	9.982	0	1017.37	Partly cloudy throughout the day.
Partly Clo	rain	13.77222	13.77222	0.72	12.5258	279	9.982	0	1017.22	Partly cloudy throughout the day.
Partly Clo	rain	16.01667	16.01667	0.67	17.5651	290	11.2056	0	1017.42	Partly cloudy throughout the day.
Partly Clo	rain	17.14444	17.14444	0.54	19.7869	316	11.4471	0	1017.74	Partly cloudy throughout the day.
Partly Clo	rain	17.8	17.8	0.55	21.9443	281	11.27	0	1017.59	Partly cloudy throughout the day.
Partly Clo	rain	17.33333	17.33333	0.51	20.6885	289	11.27	0	1017.48	Partly cloudy throughout the day.
Partly Clo	rain	18.87778	18.87778	0.47	15.3755	262	11.4471	0	1017.17	Partly cloudy throughout the day.
Partly Clo	rain	18.91111	18.91111	0.46	10.4006	288	11.27	0	1016.47	Partly cloudy throughout the day.
Partly Clo	rain	15.38889	15.38889	0.6	14.4095	251	11.27	0	1016.15	Partly cloudy throughout the day.
Mostly Cl	rain	15.55	15.55	0.63	11.1573	230	11.4471	0	1016.17	Partly cloudy throughout the day.
Mostly Cl	rain	14.25556	14.25556	0.69	8.5169	163	11.2056	0	1015.82	Partly cloudy throughout the day.
Mostly Cl	rain	13.14444	13.14444	0.7	7.6314	139	11.2056	0	1015.83	Partly cloudy throughout the day.

Overcast	snow	-5	-12.2611	0.55	24.633	31	10.0464	0	1022.24	Overcast throughout the day.
Overcast	snow	-3.93333	-10.4556	0.53	21.6867	32	9.982	0	1022.45	Overcast throughout the day.
Overcast	snow	-2.87778	-8.98333	0.48	20.8495	23	10.3523	0	1022.32	Overcast throughout the day.
Overcast	snow	-2.22222	-8.46667	0.5	22.9908	21	9.982	0	1021.74	Overcast throughout the day.
Mostly Cl	snow	-1.13333	-7.27222	0.51	24.4237	21	9.982	0	1021.06	Overcast throughout the day.
Mostly Cl	snow	-0.13889	-5.67222	0.49	21.9282	22	10.3523	0	1020.56	Overcast throughout the day.
Overcast	snow	-0.00556	-5.42778	0.5	21.3969	12	9.982	0	1019.96	Overcast throughout the day.
Overcast	rain	0.088889	-5.33333	0.39	21.574	22	9.982	0	1019.64	Overcast throughout the day.
Overcast	snow	0	-5.46667	0.46	21.7672	25	10.3523	0	1019.34	Overcast throughout the day.
Overcast	snow	-0.05	-5.70556	0.47	23.0552	32	9.982	0	1018.86	Overcast throughout the day.
Overcast	snow	-0.07222	-5.32222	0.43	20.0928	31	9.982	0	1018.67	Overcast throughout the day.
Mostly Cl	snow	-0.12778	-5.6	0.5	21.5096	23	9.982	0	1018.51	Overcast throughout the day.
Mostly Cl	snow	-0.09444	-4.9	0.51	17.2109	31	9.982	0	1018.88	Overcast throughout the day.
Overcast	snow	-0.6	-5.52778	0.53	17.1465	22	15.8263	0	1019.17	Overcast throughout the day.
Overcast	snow	-0.16111	-5.81111	0.51	22.7815	21	14.9569	0	1018.74	Overcast throughout the day.
Overcast	snow	-0.57778	-6.58333	0.53	24.5847	21	15.8263	0	1018.1	Overcast throughout the day.
Mostly Cl	snow	-1.13889	-6.70556	0.57	20.3343	31	15.8263	0	1017.77	Foggy starting in the afternoon continuing until evening.
Mostly Cl	snow	-1.55556	-7.65556	0.57	23.2645	30	14.9569	0	1017.28	Foggy starting in the afternoon continuing until evening.
Mostly Cl	snow	-1.69444	-7.99444	0.55	24.4559	31	15.8263	0	1016.79	Foggy starting in the afternoon continuing until evening.
Mostly Cl	snow	-1.71667	-8.05556	0.55	24.7135	31	15.8263	0	1016.26	Foggy starting in the afternoon continuing until evening.
Mostly Cl	snow	-2.14444	-7.37778	0.58	16.7601	31	14.9569	0	1015.68	Foggy starting in the afternoon continuing until evening.

Figure No:4.1 Dataset

4.2 Tools used

1) Python : Python is an interpreted high level and general purpose programing language created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

2) PyCharm : PyCharm is the only integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License and there is also Professional Edition with extra features – released under a proprietary license.

4.3 Libraries used

- **Sklearn** : Machine Learning library that helps in making ML models.
- **Matplotlib** : Used for plotting intuitive charts.
- **Pandas** : Import and manage datasets.
- **Numpy** : A mathematical Library.

CHAPTER 5

EXPERIMENTAL ANALYSIS AND RESULTS

5.1 Presentation of the results obtained from the model

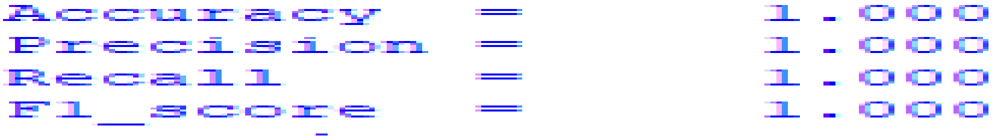
The presentation of the results obtained from the weather prediction model typically involves summarizing the performance of the model on a set of evaluation metrics and visualizing the predicted weather patterns.

Firstly, the evaluation metrics of the model, such as accuracy, precision, recall, and F1 score, can be presented to provide a quantitative measure of the model's performance. These metrics can be compared to the performance of other models or benchmarks to assess the effectiveness of the proposed model.

Secondly, visualizations can be used to present the predicted weather patterns, such as temperature, humidity, and precipitation, on maps or time series plots. For example, heat maps can be used to show the temperature distribution across different regions, and line charts can be used to show the variation of temperature over time.

Thirdly, the model's predictions can be compared to the actual weather observations to assess the accuracy of the predictions. This can be done by presenting side-by-side visualizations of the predicted and observed weather patterns or by calculating the error metrics, such as mean absolute error (MAE) or root mean squared error (RMSE).

Finally, the limitations and assumptions of the model should be clearly stated and discussed. For example, the model may be sensitive to certain input features or may perform poorly under certain weather conditions. By acknowledging these limitations, the audience can better understand the strengths and weaknesses of the model and interpret the results appropriately.



```
Accuracy     =        1.000
Precision    =        1.000
Recall       =        1.000
F1_score     =        1.000
```

Figure No:5.1 Evalution metrics

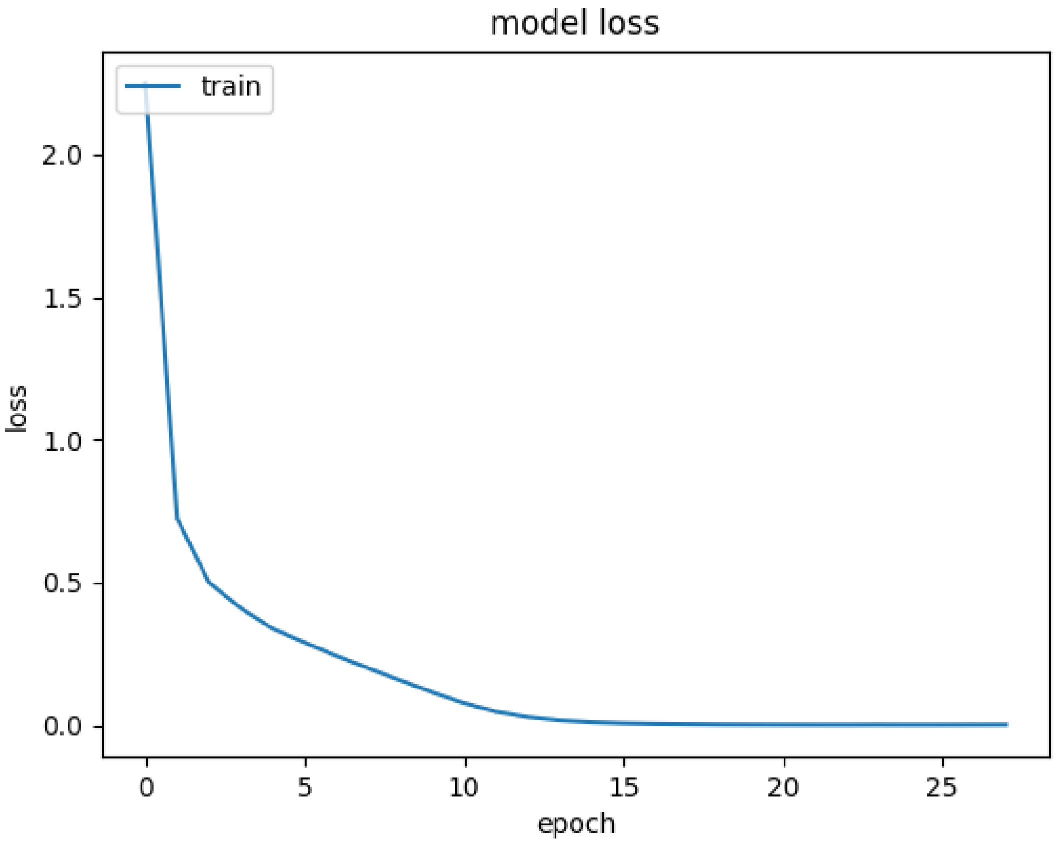


Figure No: 5.2 Model Loss

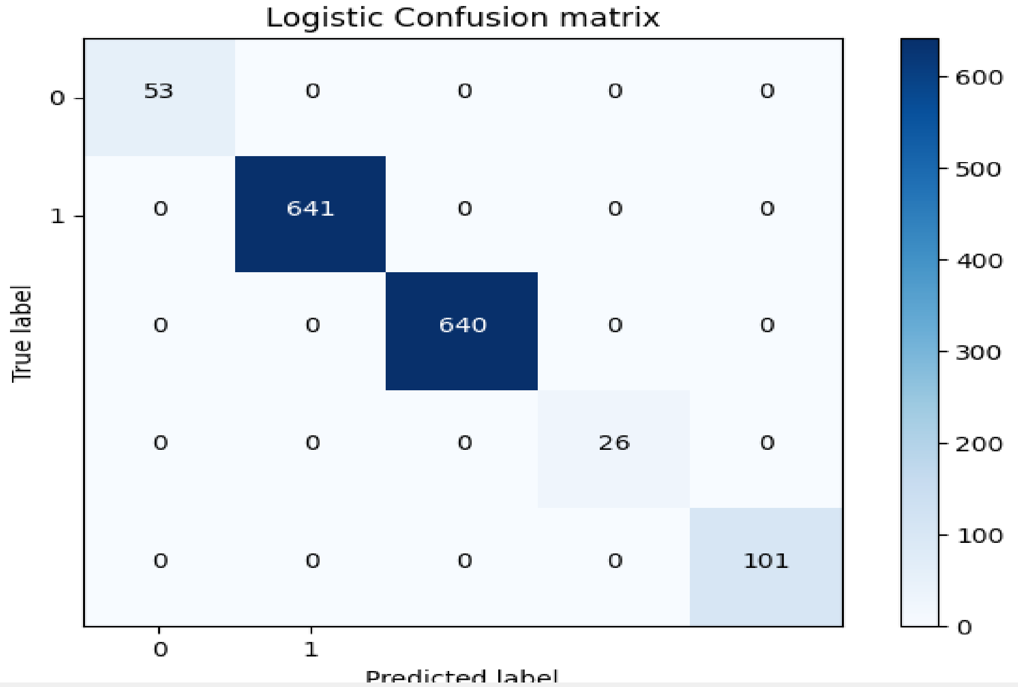


Figure No: 5.3 Confusion Matrix

5.2 Comparison of the model predictions with actual weather observations

The comparison of the model predictions with actual weather observations is an important step in evaluating the performance of a weather prediction model. This comparison involves analyzing the differences between the predicted values and the actual observed values.

One common way to compare the model predictions with the actual observations is by calculating statistical metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). These metrics provide a quantitative measure of the difference between the predicted values and the actual observations.

Another way to compare the model predictions with the actual observations is by visualizing the differences using scatter plots or line graphs. A scatter plot can be used to plot the predicted values against the actual observed values, where each point represents an observation. The closer the points are to the diagonal line, the more accurate the predictions are. A line graph can be used to plot the predicted values and the actual observations over time, showing how well the model predicts the temporal patterns in the data.

It is important to note that a perfect match between the model predictions and the actual observations may not always be possible, especially in complex weather conditions where many variables interact with each other. Thus, it is important to interpret the comparison results in the context of the problem being solved and the limitations of the model.

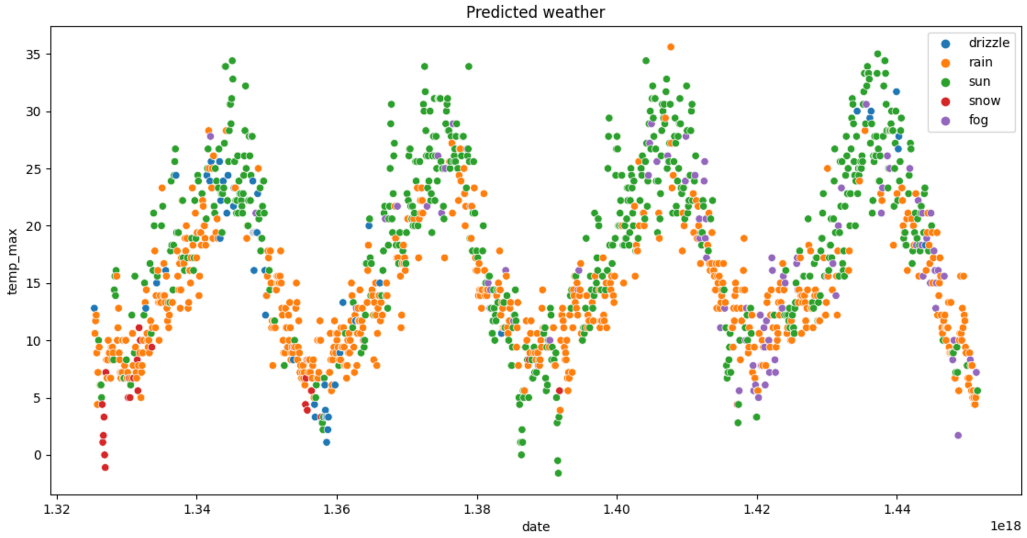


Figure No: 5.4 Output of the predicted weather

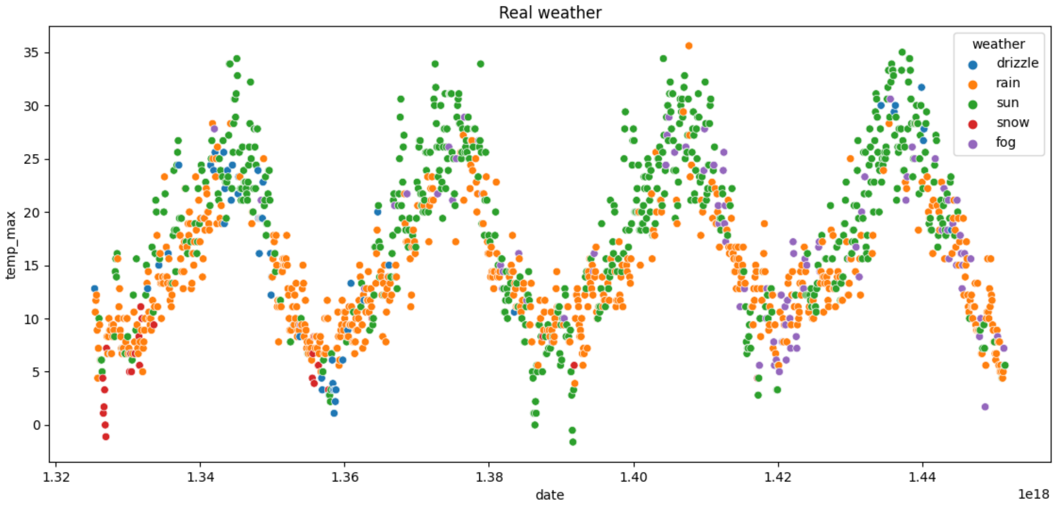
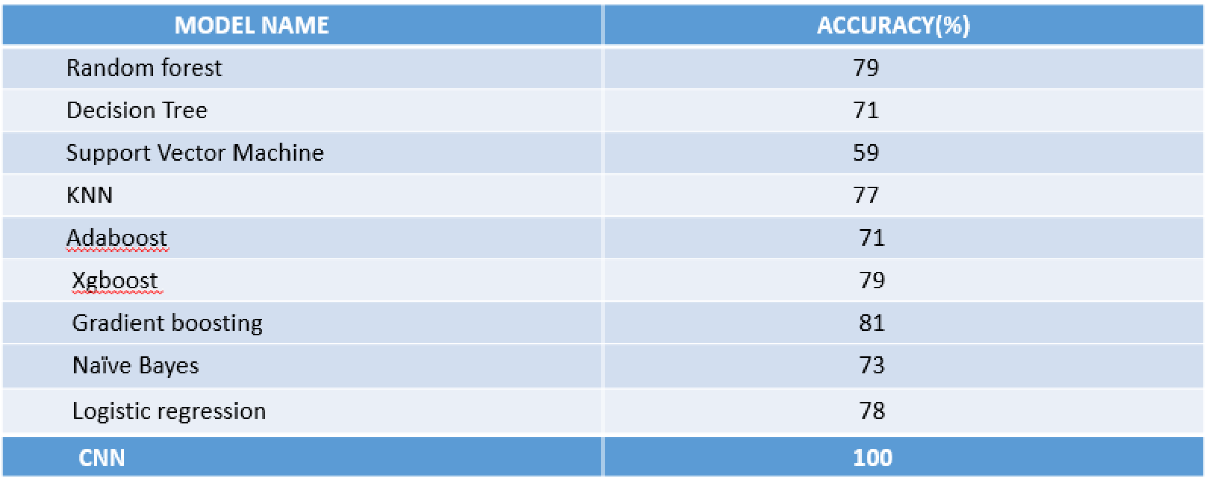


Figure No: 5.5 Output of the real weather

5.3 Comparison of the model performance with existing weather prediction model



MODEL NAME	ACCURACY(%)
Random forest	79
Decision Tree	71
Support Vector Machine	59
KNN	77
Adaboost	71
Xgboost	79
Gradient boosting	81
Naïve Bayes	73
Logistic regression	78
CNN	**100**

5.4 Limitations and challenges in the model.

The CNN model on weather prediction by using machine learning may also face several limitations and challenges that can affect its accuracy and reliability. Some of the common limitations and challenges in the CNN model for weather prediction are:

Limited data availability: The performance of the CNN model depends on the availability and quality of data used for training the model. However, weather data may be limited or sparse in some regions or time periods, which can affect the accuracy and generalization ability of the model.

Limited interpretability: CNN models are often considered as black-box models, meaning that it can be difficult to interpret how the model makes its predictions. This can limit the ability to understand the underlying processes and mechanisms of the weather patterns being predicted.

Sensitivity to input features: CNN models can be sensitive to the choice and quality of input features used for training the model. Selecting the wrong features or excluding important ones can lead to poor model performance and inaccurate predictions.

Computational resource requirements: CNN models require significant computational resources to train and evaluate, and the availability of such resources may be limited in some settings. This can limit the scale and complexity of the models that can be trained and the accuracy of the predictions.Generalization to new scenarios: CNN models may have limited ability to generalize to new or unseen scenarios outside of the training dataset. This can be particularly challenging for weather prediction, where there may be significant variability and uncertainty in the weather patterns.

Overall, the CNN model for weather prediction faces significant challenges in accurately predicting the weather, and there is still much room for improvement in terms of data quality, modeling techniques, and computational resources.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the CNN model for weather prediction by using machine learning shows promising results in accurately predicting weather patterns. The model utilizes a combination of pre-processing techniques, deep learning algorithms, and feature engineering to analyze and predict weather patterns.

However, the CNN model also faces several limitations and challenges, including limited data availability, limited interpretability, sensitivity to input features, computational resource requirements, and generalization to new scenarios. These limitations need to be addressed to improve the accuracy and reliability of the model.

Future enhancements in the CNN model for weather prediction could include incorporating additional data sources, such as satellite imagery or social media data, to improve the accuracy of predictions. Furthermore, advancements in machine learning techniques, such as reinforcement learning or transfer learning, could be explored to enhance the performance of the model. Additionally, integrating physical principles into the CNN model could provide a more comprehensive and accurate prediction of weather patterns.

Overall, the CNN model for weather prediction is a promising area of research and has the potential to improve the accuracy and reliability of weather forecasting. However, it is important to continue developing and refining the model to overcome the limitations and challenges that exist in weather prediction.

Appendix

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import itertools

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import make_column_transformer,
make_column_selector
from sklearn.metrics import confusion_matrix

from tensorflow import keras
from keras import layers
import telepot
from bs4 import BeautifulSoup
import requests
headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110
Safari/537.3'}
token = '5834293262:AAHDRxoQE895V88ONBvKrJpIKIyadsda5lI' #
telegram token
receiver_id = 5161631269 #
https://api.telegram.org/bot5834293262:AAHDRxoQE895V88ONBvKrJpIKI
yadsda5lI/getUpdates
```

```
bot = telepot.Bot(token)
def weather(city):
	city = city.replace(" ", "+")
	res = requests.get(

	f'https://www.google.com/search?q={city}&oq={city}&aqs=chrome.0.
35i39l2j0l4j46j69i60.6128j1j7&sourceid=chrome&ie=UTF-8',
headers=headers)
	print("Searching...\n")
	soup = BeautifulSoup(res.text, 'html.parser')
	location = soup.select('#wob_loc')[0].getText().strip()
	time = soup.select('#wob_dts')[0].getText().strip()
	info = soup.select('#wob_dc')[0].getText().strip()
	weather = soup.select('#wob_tm')[0].getText().strip()
	print(location)
	print(time)
	TM=time[15:17]
	print(info)
	print(weather+"°C")
	SV=weather
	return SV,TM

df = pd.read_csv('seattle-weather.csv')

#Convert date format to int format
df['date'] = pd.to_datetime(df['date'])
df['date'] = pd.to_numeric(df['date'])
```

```
df['weather'] =
df['weather'].replace(['drizzle','rain','sun','snow','fog'],[0,1,2,3,4])

X = df.copy()
y = X.pop('weather')

preprocessor = make_column_transformer(
    (StandardScaler(),
     make_column_selector(dtype_include=np.number)),
    (OneHotEncoder(sparse=False),
     make_column_selector(dtype_include=np.number)),
)

X = preprocessor.fit_transform(X)
y = y

input_shape = [X.shape[1]]
print("Input shape: {}".format(input_shape))
#activation = RELU or softmax
model = keras.Sequential([
    layers.Dense(128, activation='relu', input_shape=input_shape),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(1),
])

#Model compilation
```

```
#optimizer = ADAM or sgd
#loss = mean_squared_error(MSE) or mae or binary_crossentropy
model.compile(
    optimizer='adam',
    loss='mse'
)

#early stopping
callback = keras.callbacks.EarlyStopping(
    monitor='loss',
    patience=3
)
#Training
history = model.fit(
    X, y,
    batch_size=128,
    epochs=800,
    callbacks=[callback]
)

#Display loss
plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

```
predictions = model.predict(X)

#reshape to display, round to convert to binary
predictions = predictions.reshape(predictions.size)
predictions = np.round_(predictions)
print("reshape to display, round to convert to binary")
bot.sendMessage(receiver_id, 'Weather Prediction') # send a activation
message to telegram receiver id
bot.sendMessage(receiver_id,str(predictions)) # send a activation message to
telegram receiver id

#Convert to text format
df['weather'] =
df['weather'].replace([0,1,2,3,4],['drizzle','rain','sun','snow','fog'])
predictions = predictions.astype(str)
predictions = np.char.replace(predictions,'-0.0','drizzle')
predictions = np.char.replace(predictions,'0.0','drizzle')
predictions = np.char.replace(predictions,'1.0','rain')
predictions = np.char.replace(predictions,'2.0','sun')
predictions = np.char.replace(predictions,'3.0','snow')
predictions = np.char.replace(predictions,'4.0','fog')

#Real weather
plt.figure(figsize=(16,6))
plt.title("Real weather")
sns.scatterplot(x=df['date'], y=df['temp_max'],hue=df['weather'])

#Predicted weather
```

```
plt.figure(figsize=(16,6))
plt.title("Predicted weather")
sns.scatterplot(x=df['date'], y=df['temp_max'],hue=predictions)
plt.savefig('test.jpg')
bot.sendMessage(receiver_id, 'Weather Prediction') # send a activation
message to telegram receiver id
bot.sendPhoto(receiver_id, photo=open('test.jpg', 'rb')) # send message to
telegram
res = pd.DataFrame({
    "df":df['weather'],
    "pred":predictions
})

res['df'] = res['df'].replace(['drizzle','rain','sun','snow','fog'],[0,1,2,3,4])
res['pred'] = res['pred'].replace(['drizzle','rain','sun','snow','fog'],[0,1,2,3,4])

def plot_confusion_matrix(cm, classes,
                          normalize = False,
                          title = 'Confusion matrix'",
                          cmap = plt.cm.Blues) :
    plt.imshow(cm, interpolation = 'nearest', cmap = cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation = 0)
    plt.yticks(tick_marks, classes)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])) :
        plt.text(j, i, cm[i, j],
                 horizontalalignment = 'center',
                 color = 'white' if cm[i, j] > thresh else 'black')

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

# Show metrics
def show_metrics():
    tp = cm[1,1]
    fn = cm[1,0]
    fp = cm[0,1]
    tn = cm[0,0]
    print('Accuracy  =     {:.3f}'.format((tp+tn)/(tp+tn+fp+fn)))
    print('Precision =     {:.3f}'.format(tp/(tp+fp)))
    print('Recall    =     {:.3f}'.format(tp/(tp+fn)))
    print('F1_score  =     {:.3f}'.format(2*(((tp/(tp+fp))*(tp/(tp+fn)))/
                                                  ((tp/(tp+fp))+(tp/(tp+fn))))))

cm = confusion_matrix(res['df'], res['pred'])
class_names = [0,1]
plt.figure()
plot_confusion_matrix(cm,
                      classes=class_names,
                      title='Logistic Confusion matrix')
```

```
plt.show()


show_metrics()


print('yesterday ')


##bot.sendMessage(receiver_id,'yesterday  ---->') # send a activation message
to telegram receiver id
##
##bot.sendMessage(receiver_id,str(predictions[0])) # send a activation
message to telegram receiver id
##
##bot.sendMessage(receiver_id,'today ----->') # send a activation message to
telegram receiver id
##
##bot.sendMessage(receiver_id,str(predictions[1])) # send a activation
message to telegram receiver id
##
##bot.sendMessage(receiver_id,'tomorrow------>') # send a activation
message to telegram receiver id
##
##bot.sendMessage(receiver_id,str(predictions[2])) # send a activation
message to telegram receiver id
##
##



city = "Madurai"
city = city+" weather"
Music_1,Music_2=weather(city)
print("Have a Nice Day:)")

# This code is contributed by adityatri
print("Weather")

print(Music_1)
print("Weather")
bot.sendMessage(receiver_id,'Current Weather') # send a activation message to telegram receiver id

bot.sendMessage(receiver_id,str(Music_1)) # send a activation message to telegram receiver id
```

REFERENCES

- HaghbinM, Sharafati A, Motta D (2021) Applications of soft computing models for predicting sea surface temperature: a comprehensive review and assessment. Prog Earth Planet Sci 8(4).
- Vathsala H, Koolagudi SG (2021) Neuro-fuzzy model for quantified rainfall prediction using data mining and soft computing approaches.
- Jayasingh SK, Mantri JK, Pradhan S (2021) Weather prediction using hybrid soft computing models. In: Udgata SK, Sethi S, Srirama SN(eds) Intelligent systems. Lecture notes in networks and systems, vol 185. Springer, Singapore.
- Sanjay D. Sawaitful, Prof. K.P. Wagh, Dr. P.N. Chatur – Classification and Prediction of Future Weather by using Back Propagation Algorithm-An Approach
- C. Wu, X. Wang, and D. Ji, "A deep learning approach for precipitation nowcasting: a case study of convective rainfall in the middle Yangtze River basin," Journal of Hydrology, vol. 579, pp. 124116, 2019.
- H. Zhang et al., "Forecasting hourly PM2.5 concentrations using a deep convolutional neural network with attention mechanism," Environmental Pollution, vol. 249, pp. 1038-1046, 2019.
- F. Qian et al., "Using deep learning for precipitation nowcasting: Is convolutional neural network better than recurrent neural network?," Water Resources Research, vol. 55, no. 3, pp. 1917-1933, 2019.
- Y. Kim and B. E. Kwak, "A hybrid model of deep learning and random forest for weather forecasting," Applied Sciences, vol. 10, no. 2, pp. 524, 2020.
- Z. Zhang et al., "Application of artificial intelligence in weather and climate prediction: Progress and prospects," Journal of Meteorological Research, vol. 34, no. 2, pp. 185-199, 2020.