**Weather Forecasting Using Data Mining Techniques**

**TABLE OF CONTENTS**

**CONTENTS**

**CERTIFICATE**

**DECLARATION**

**ACKNOWLEDGEMENT**

**ABSTRACT**

**INDEX**

1. **INTRODUCTION**
   1. PURPOSE
   2. SCOPE
   3. DEFINITIONS, ACRONYMS, ABBREVIATIONS
   4. TECHNOLOGIES TO BE USED
   5. REFERENCES
   6. OVERVIEW
2. **LITERATURE REVIEW**
3. **PROBLEM DEFINITION**
4. **REQUIREMENT SPECIFICATIONS**
   1. PRODUCT PERSPECTIVE
   2. SOFTWARE REQUIREMENTS
   3. HARDWARE REQUIREMENTS
   4. CONSTRAINTS
   5. PRODUCT FUNCTION
   6. SYSTEM MODEL
   7. PROCESS MODEL
5. **SYSTEM ANALYSIS**
6. **SYSTEM DESIGN**
   1. LOGICAL DESIGN
   2. PHYSICAL DESIGN
   3. UML DIAGRAMS
      1. USE CASE DIAGRAM
      2. SEQUENCE DIAGRAM
      3. CLASS DIAGRAM
      4. COLLABORATION DIAGRAM
7. **CODING**
8. **TESTING**
9. **CONCLUSION AND FUTURE SCOPE**
10. **REFERENCES**
11. **APPPENDIX**

**ABSTRACT**

Weather forecasting is a vital application in meteorology and has been one of the most scientifically and technologically challenging problems around the world in the last century. Weather is a continuous, data-intensive, multidimensional, dynamic process that makes weather forecasting a formidable challenge. The rate of acquiring data is exploding and managing such data so as to infer useful knowledge that can be put to use, is becoming important. Data Mining is one such technology that is employed in inferring useful knowledge that can be put to use from a vast amount of data. Data mining is an interesting field of computer science that can be used for various applications. This project aims at developing a weather information system as a web service that can be used by any type of application and uses the prediction technique of data mining for weather forecasting.

1. **INTRODUCTION**

Data mining, also called Knowledge Discovery in Databases (KDD), is the process of extracting or mining knowledge from large amount of data. In other words Data mining is the efficient discovery of valuable, non-obvious, novel and potentially useful information from a large collection of data. It extracts hidden predictive information from large databases, thus is a powerful new technology with great potential to help in analysis of data and for decision making. Data mining functionalities are used to specify the kind of patterns to be found in general data mining tasks. In general data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions. This techniques are often more powerful, flexible, and efficient for exploratory analysis than the statistical techniques. The most commonly used implementations for the techniques in data mining are: Artificial Neural Networks, Genetic Algorithms, Rule Induction, Nearest Neighbour method, Memory-Based Reasoning, Logistic Regression, Discriminant Analysis and Decision Trees.

Data mining is an interesting technique that can be implemented to many areas. Some of the applications of data mining include discovery of interesting patterns, clustering of data based on parameters and prediction of results by using the existing data. There are diverse techniques and algorithms available in data mining that can be implemented for various applications. Meteorological data mining is a form of Data mining concerned with finding hidden patterns inside largely available meteorological data, so that the information retrieved can be transformed into usable knowledge. Useful knowledge can play important role in understanding the climate variability and climate prediction. This understanding can be used to support many important sectors that are affected by climate like agriculture, water resources and tourism. To make an accurate prediction is one of the major challenges facing meteorologist all over the world. Weather is one of the meteorological data that is rich by important knowledge.

Weather forecasting is a prediction of what the weather will be like in future, it had been invented many years ago. A weather forecast involves five steps: observation, collection and transformation of data, plotting of weather data, analysis of data and extrapolation to find the future state of the atmosphere, and prediction of particular variables. There are various classification methods that can be used, like Naïve Bayes classifier, Decision Trees, Artificial Neural Networks and Support Vector Machines which come under the category of supervised methods, whereas the unsupervised method is an adaptation of the K-means clustering method.

Accurate prediction of weather parameters is a difficult task due to the dynamic nature of atmosphere. Various techniques like linear regression, auto regression, Multi Layer Perceptron, Radial Basis Function networks are applied to predict atmospheric parameters like temperature, wind speed, rainfall, meteorological pollution etc. It was found that the non linear operator equations governing the atmospheric system are the ones who can better understand the dynamics of atmosphere. In the recent past many forecast methods have been developed using Artificial Neural Networks. However systems developed using neural network model suffer from certain drawbacks.

In this work, we make use of Support Vector Machine algorithm for forecasting the weather. Support Vector Machines are often superior to ANNs because they avoid two major weaknesses of ANNs:

(1) ANNs often converge on local minima rather than global minima, meaning that they are essentially "missing the big picture" sometimes (or missing the forest for the trees)

(2) ANNs often over fit if training goes on too long, meaning that for any given pattern, an ANN might start to consider the noise as part of the pattern.

SVMs don't suffer from either of these two problems.

Support vector machines (SVMs) use a linear model to implement nonlinear class boundaries through some nonlinear mapping input vectors into a high-dimensional feature space. The linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyper plane (OSH) is constructed. Thus, SVM is known as the algorithm that finds a special kind of linear model, the maximum margin hyper plane. The maximum margin hyper plane gives the maximum separation between decision classes. The training examples that are closest to the maximum margin hyper plane are called support vectors. A version of SVM for [regression](http://en.wikipedia.org/wiki/Regression_analysis) was proposed in 1996 by [Vladimir N. Vapnik](http://en.wikipedia.org/wiki/Vladimir_N._Vapnik), Harris Drucker, Christopher J. C. Burges, Linda Kaufman and Alexander J. Smola. This method is called [support vector regression](http://en.wikipedia.org/w/index.php?title=Support_vector_regression&action=edit&redlink=1) (SVR), used for atmospheric temperature prediction.

SVM models can be obtained using a SVM toolkit in MATLAB or by LIBSVM for MATLAB or by simply using a WEKA tool. This project shows its implementation using MATLAB technique. LIBSVMis integrated software for support vector classification, regression and distribution. It supports multi-class classification as well.LIBSVM provides a simple interface where users can easily link it with their own programs. MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

In this approach, we use .NET web service, in order to input the weather parameters we ought to classify and as well display the classified results. It serves as a front end for using the web service. A Web service is a method of communication between two electronic devices over World Wide Web. A Web service is a software function provided at a network address over the web or the cloud. Roughly speaking, Web Services are applications that can be published, located, and invoked across the Internet. Typical examples include:

* Getting stock price information.
* Obtaining weather reports.
* Making flight reservations.
  1. **Purpose**

To discover and understand the dynamic phenomena of weather, as well make an accurate prediction have been integral components of scientific investigation world-wide. The purpose of this project is to extract the patterns for day-to-day weather prediction from historical weather data using data mining. Furthermore, the projected solution was cost effective and requires minimal intervention of weather equipment. Requirements were gathered based on the literature review done by analyzing similar products and current weather prediction methodologies.

* 1. **Scope**
* After collecting, analysing the daily weather observations, temperature, pressure, dew point and humidity were selected for predicting a weather event to be Clear, Foggy, Scattered Clods, Thunderstorm, Partly Cloudy, Rain and Rain-Thunderstorm.
* The project to be made must have the scope for expanding or scope for switching to a new technology later, if needed.
* The major scope of the project is to determine the weather of a particular area, basing upon the already recorded data.
  1. **Definitions, acronyms, abbreviations**

**UML :** Unified Modelling Language

**SVM :** Support Vector Machine

**C# :** C sharp

* 1. **Technologies to be used:**
* Visual C#
* Microsoft Visual Studio 2010
* MATLAB
* Rational Rose Enterprise Edition: version 3
  1. **References:**

**Book and Paper References**

* Data Mining Concepts and Techniques Jiawei Han and Micheline Kamber Morgan Kaufman Publications.
* A Service Oriented Architecture for Weather Forecasting Using Data Mining Mrs. C. Beulah Christalin Latha
* Y.Radhika and M.Shashi, Atmospheric Temperature Prediction using Support Vector Machines, International Journal of Computer Theory and Engineering, Vol. 1, No.
* Hartwig Gunzer, Introduction to Web Services, ZDNet, March 2002.

**Web References**

* [www.google.co.in](http://www.google.co.in)
* [www.wikipedia.org](http://www.wikipedia.org)
  1. **Overview:**
* **Overall Description:**

The Day-by-day variations in a given area constitute the weather, whereas climate is the long-term synthesis of such variations. Weather is measured by thermometers, rain gauges, barometers, and other instruments, but the study of climate relies on statistics. Nowadays, such statistics are handled efficiently by computers. A simple, long-term summary of weather changes, however, is still not a true picture of climate. To obtain this requires the analysis of daily, monthly, and yearly patterns.

Climate change today is synonymous with anthropogenic global warming. Evidence for climatic change is taken from a variety of sources that can be used to reconstruct past climates. Data mining is the field of discovering novel and potentially useful information from large amounts of data. In contrast to standard statistical methods, data mining techniques search for interesting information without demanding a priori hypotheses, the kind of patterns that can be discovered depend upon the data mining tasks employed.

**Materials and Methods generally used are**:

Data Collection

The data used for this work was collected from website of University Of Cambridge.

Data Cleaning

In this stage, a consistent format for the data model was developed which took care of missing data, finding duplicated data, and weeding out of bad data. Finally, the cleaned data were transformed into a format suitable for data mining.

Data Analysis and Mining Stage

In this stage, the Support Vector Machine algorithm is implemented; SVM models are created to predict the weather, using the current or already present data.

* **Specific Requirements:**

1. The essential requirement is the weather data on day-to-day basis. The attributes to be collected are mainly temperature, humidity, pressure, and dew-point.
2. This data is pre-processed using some data cleaning techniques, and thus the missing values are dealt.
3. The data now is ready for the analysis process. This is done by constructing SVM models, using MATLAB.
4. This data is classified into several attributes like :

* Rain
* Rain, Thunderstorm
* Haze
* Haze, Scattered Clouds
* Fog, Rain, Thunderstorm
* Fog, Rain, Hail, Thunderstorm
* Thunderstorm
* Haze, Partly Cloudy
* Fog
* Mist, Rain
* Haze, Scattered Cloud , Mist
* Haze, Mist
* Haze, Mist, Partly Cloudy

These classified attributes as an output help in predicting the weather depending on the four attributes taken as an input data.

1. The front end interface is created using .NET, which helps to view the output attributes or the climatic condition we have predicted for the set of input data.

**2. LITERATURE REVIEW:**

Indian Meteorological Department (IMD) has progressively expanded its infrastructure for meteorological observations, communications, forecasting and weather services and it has concurrently contributed to scientific growth. Denis Riordan and Bjarne K Hansen (2002) stated that Case-based reasoning is emerging as a leading methodology for the application of artificial intelligence. They described an investigation into the application of case-based reasoning in airport weather forecasting. Knowledge about temporal features that human forecasters use to construct analogous climatic scenarios is encoded in a fuzzy similarity measure. A fuzzy case-based system for weather prediction similarity measure is used to locate the k-nearest neighbours from the historical database. These nearest neighbours are in turn adapted to produce values for the forecast parameters, which increase the accuracy of predictions. This approach of fuzzification was further developed and a new method for classification was also implemented using Artificial Neural Networks.

Imran Maqsood et.al, (2004) have proposed an ensemble of neural networks approach for weather forecasting. This study presented the weather forecasting in southern Saskatchewan; Canada. The proposed method used back then, for weather forecasting was advantageous over other techniques like linear combination. Generally, the output of an ensemble is a weighted sum, which is weight-fixed, with the weights being determined from the training or validation data. In the proposed approach, weights are determined dynamically from the respective certainties of the outputs. Further, the use of ANN’s in this sector was implemented.

Dr. S. Santhosh Baboo and I.Kadar Shereef(2011) also, have stated that temperature warnings are important forecasts because they are used to protect life and property. Temperature forecasts are made by collecting quantitative data about the current state of the atmosphere. They have present a neural network-based algorithm for predicting the temperature. The Neural Networks package supports different types of training or learning algorithms. One such algorithm is Back Propagation Neural Network (BPN) technique.The main advantage of the BPN neural network method is that it can fairly approximate a large class of functions. This method is more efficient than numerical differentiation.

Several steps followed to predict the temperature were:

a. Data collection (atmospheric pressure, temperature,

wind speed and direction, humidity, precipitation) as inputs.

b. Data assimilation and analysis

c. Numerical weather prediction

d. Model output post processing

Through the implementation of this system, they illustrated how an intelligent system can be efficiently integrated with a neural network prediction model to predict the temperature. The algorithm improved convergence and damped the oscillations. The results obtained were compared with actual working of meteorological department and those results confirm that their model had the potential for successful application to temperature forecasting. Real time processing of weather data indicate that the BPN based weather forecast have shown improvement not only over guidance forecasts from numerical models, but over official local weather service forecasts as well.

Folorunsho Olaiya, Adesesan Barnabas Adeyemo (2012) have investigated the usage of data mining techniques in forecasting maximum temperature, rainfall, evaporation and wind speed. This was carried out using Artificial Neural Network and Decision Tree algorithms and meteorological data collected between 2000 and 2009 from the city of Ibadan, Nigeria. A data model for the meteorological data was developed and this was used to train the classifier algorithms. The performances of these algorithms were compared using standard performance metrics, and the algorithm which gave the best results used to generate classification rules for the mean weather variables. A predictive Neural Network model was also developed for the weather prediction program and the results compared with actual weather data for the predicted periods. The results show that given enough case data, Data Mining techniques can be used for weather forecasting and climate change studies.

An improved approach over the neural networks was implemented by Huey-Ru Wu et.al, which was, Support Vector Machine on Analysis of Temperature and Precipitation Data. Here, they have introduced the usage of a quite powerful classification tool in data mining area to find a way to deal with the weather prediction. They have picked out observation data of two seasons from two weather stations, and used the data of summer and winter from year 1995 (June, 1995) to 2008 (February, 2009) was used. The selected data from June to August was used as summer data and December to February of next year as winter data. They then used the classification tool, support vector machine, for analysis.

Finally, the results were analyzed and discussed. The overall accuracy was more than 90% in average; and was more than 80% even in the worst case. This showed the possibility and power of the SVM tools. Classification and Prediction analysis in data mining has been a vast area of research and implementation, since the past years. Weather Forecast is one of the important areas where classification technique plays a major role.

**3. PROBLEM DEFINITION:**

Weather is one of the meteorological data that is rich by important knowledge. Extreme weather becomes an important topic in recent years, while how to predict the occurrence of extreme weather is still a tough task. Weather prediction has drawn a lot of results in previous literature, and recent studies have shown that machine learning techniques achieved better performance than traditional statistical ones. Data Mining is one such area which bridges the gap between generation of data and understanding of the data. In this project, we investigate the use of data mining techniques in forecasting the weather. Predicting various weather events like haze, mist, cloud, rain, thunderstorm etc. for a day, by collecting temperature, pressure, humidity and dew point. Pre-process the data, to handle with all the unknown and missing values. On the clean data set, apply the classification technique, and implement it by constructing Support Vector Machine models for the Weather data collected from University Of Cambridge for a period of one year, in order to forecast the weather.

**4. REQUIREMENT SPECIFICATIONS:**

**4.1 Product Perspective**:

To provide a system for predicting weather events using support vector machines (SVM models) depending on various attributes like temperature, humidity, dew point and pressure.

**4.2 Software Requirements:**

* Visual C#
* Microsoft Visual Studio 2010
* MATLAB

**4.3 Hardware Requirements:**

* RAM : 1GB
* Hard Disk : 160 GB
* Visual display unit

**4.4 Constraints:**

* As weather is highly ambiguous, it is very difficult to predict exact weather events for a day depending on learning data-set. Weather keeps on changing every year due to global warming and hence the model may not predict the exact weather events in the coming years also

* Graphical User Interface (GUI) is only in English

**4.5 Product Functions:**

* Attributes- mean temperature, humidity, pressure, dew point.
* Class labels- {Rain}, {Rain, thunderstorm}, {Haze}, {Haze, scattered clouds}, {Fog, Rain, thunderstorm}, {Fog, Rain, Hail, thunderstorm}, {Thunderstorm}, {Haze, partly cloudy}, {Fog}, {Mist, Rain}, {Haze, scattered clouds, Mist}, {Haze, Mist}, {Haze, Mist, partly cloudy}, {Mist, Fog, shallow fog}.
* User can know the weather events of the day.

**4.6 System Model:**

SVM MODELS

DATA ANALYSIS

Pressure

Dew Point

Humidity

Temperature

COLLECT DATA

HANDLE

MISSING VALUES

PREPROCESSED DATA SET

ESTIMATION AND PREDICTION

FIG 1 **SYSTEM MODEL**

**4.7 Process Model:**

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. Also it does not define the process to go back to the previous phase to handle changes in requirement.

The waterfall model is the earliest approach that was used for software development. Initially, most projects followed the waterfall approach because they did not focus on changing requirements.

**WATERFALL APPROACH PHASES**

1. **Conception:**

Triggers when a problem is perceived. This phase involves identifying goals to be achieved after the problem is solved, estimating benefits in the new system over the current system, and identifying other areas that are affected by the solution. This phase also involves and developing the business case for the project. A business case provides the information that a manger needs to decide whether to support a proposed project, before resources are committed to its development.

1. **Initiation:**

Involves a macro level study of the customer requirements. This phase also involves defining alternative solutions to the customer requirements and cost-benefit justification of these alternatives.

1. **Analysis:**

Involves carrying out a detailed study of the customer requirements and arriving at the exact requirements of the proposed system. This phase involves freezing the requirements before the design phase begins.

DESIGN

DEPLOYMENT

TESTING

ANALYSIS

CONSTRUCTION

INITIATION

CONCEPTION

FIG 2 **WATERFALL MODEL**

1. **Design:**

Involves translating the identified requirements into a logical structure, called design that can be implemented in a programming logic.

1. **Construction:**

Involves integrating and testing all the modules developed in the previous phase.

1. **Testing:**

Involves integrating and testing all the modules developed in the previous phase as a complete system.

1. **Deployment:**

Involves converting the new system design into operation. This may involve implementing the software system and training the operating staff before the software system is functional.

The waterfall approach assumes that requirements are stable and frozen across the project plan. However, this is usually not true in case of large projects where requirements may evolve across the development process.

**IMPLEMENTATION IN OUR PROJECT**

In **conception phase**, we studied the problem and collected the information of the resources from the website of University Of Cambridge.

In **Initiation phase**,for our implementation we installed Microsoft Visual Studio. For development purpose we have used C#.

In **Analysis phase**,we completely studied the proposed system and planned how to implement this proposed system by using Support Vector Machine models.

In **Designing phase**, we implemented the designing part of the project using visual c# in Microsoft Visual Studio 10.

In **Construction and testing phase**, we constructed and tested the system with the test sets and finally integrated them.

In **Deployment phase**,we deployed the application in the proposed system and verified by different test cases.

**5. SYSTEM ANALYSIS:**

Extreme weather has become an important topic in recent years, while how to predict the occurrence of extreme weather is still a tough task. We tried to introduce a quite powerful classification tool in data mining area to find a way to deal with this task. The data collected is analyzed, predicted using the SVM models. We can make use of Support Vector Machine (SVM) when our data has exactly two classes. A SVM classifies data by finding the best hyper plane that separates all data points of one class from those of the other class. The best hyper plane for a SVM means the one with the largest margin between the two classes. Where, margin is the maximum width of the slab parallel to the hyper plane that has no interior data points. The Support Vectors are the data points that are closest to the separating hyper plane; these points are on the boundary of the slab. Here, we implement the SVM models through MATLAB which allows matrix multiplications, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN.

In this work, we make use of .NET environment with C# programming to create the front end, which acts as a medium to display the data in form of the classified attributes. C#(pronounced see sharp) is a multi-paradigm programming language, encompassing strong typing, imperative and declarative, functional and component-oriented programming disciplines. It aims at combining the high productivity of Visual Basic, elegance of Java and raw power of Microsoft Visual Studio 2010 to develop the front end of the project. It was developed by Microsoft within its .NET initiative. C# is intended to be suitable for writing applications for both hosted and embedded systems. C# applications in general cannot run without the .NET runtime. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

**6. SYSTEM DESIGN:**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One can see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations.

**6.1 Logical Design:**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system.

**6.2 Physical Design:**

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is inputted into a system, how it is verified or authenticated, how it is processed, and how it is displayed as output. Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

**6.3 UML Diagrams:**

**6.3.1 Use-Case Diagrams:**

* **Terms and Concepts:**

A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. You apply use case diagrams to illustrate the static use case view of a system. Use case diagrams are especially important in organizing and modelling the behaviours of a system.

Use case diagrams commonly contain:

* Use cases
* Actors
* Dependency, generalization, and association relationships

**Actors:**

An actor represents a coherent set of roles that users of use cases play when interacting with these use cases. Typically, an actor represents a role that a human, a hardware device, or even another system plays with a system.

**Use cases:**

A use case describes a set of sequences, in which each sequence represents the interaction of the things outside the system (its actors) with the system itself (and its key abstractions). These behaviours are in effect system-level functions that you use to visualize, specify, construct, and document the intended behaviour of your system during requirements capture and analysis.

**Dependency:**

A dependency is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing).

**Generalization:**

Generalization is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).

**Association:**

An association is a structural relationship that describes a set of links; a link is a connection among objects.

* **Use case Scenario:**

A use case diagram describes event sequences for an actor to use the system. It is a narrative description of process. A use case is normally actor or event based. An actor will begin a process or an event will happen that the system must respond to.



**Description for use case diagram:**

* The User collects the required weather data from the database.
* He pre-processes the data to deal with the missing values, if any.
* The user opens the application.
* He enters the parameters such as temperature, pressure, humidity, dew point.
* Analysis of the data is done by creating SVM models.
* The weather is predicted.
* The User is responsible to maintain the application and the processes included during creation of SVM models.

**Interaction Diagram:**

An interaction diagram depicts the sequence of actions that occur in a system. It captures the invocation of methods in each object and order in which invocation occurs. It is used to represent the behaviour of a system.

**6.3.2 SEQUENCE DIAGRAM**

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Consider the objects that live in the context of a system, subsystem, operation or class. Consider also the objects and roles that participate in a use case or collaboration. To model a flow of control that winds through these objects and roles, we use an interaction diagram; to emphasize the passing of messages as they unfold over time, we use a sequence diagram, a kind of interaction diagram.

****

**Description for sequence diagram:**

* User collects the weather data from the database.
* Database returns the data and the User retrieves it.
* User pre-processes the data.
* He then opens the application.
* Enters the parameters such as temperature, pressure, dew point, humidity, in the application.
* The data is analysed with the help of SVM models.
* Application generates the output and thus the weather is predicted.

**6.3.3 Class Diagram:**

This is a type of static structure diagram that describes the structure of a system by showing the system's [classes](http://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects. Classes represent an abstraction of entities with common characteristics. Association, Generalization, Dependency etc., represent the relationships between classes. It provides an overview of the target system by describing the objects and classes inside the system and the relationships between them.

Classes are represented with boxes which contain three parts:

* The top part contains the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.



**Description for Class diagram:**

* In total, eight classes are considered here, they are User, Database, Application, Temperature, Pressure, Humidity, Dew point, Result.
* User class has operations like collect (), open (), input (), analyse (). This class is in association with every other class.
* Application class ‘has an association with each of the other classes, except the database class. It has, close (), clear (), generate () as its operations.
* Database class has the attributes temp, pressure, dewpt, humidity, result and operation as return\_data ().
* Each of the classes, Temperature, Humidity, Dew Point, Pressure, and Result are in aggregation to the database class and in association to the Application class.
* Temperatures, Pressure, Humidity, Dew point, Result classes have attributes temp, pressure, humidity, dewpt, result respectively and an operation called analyse (), for each.
* User collects the data from the database, cleans the data, he then opens the application, inputs the parameters and analyses the data in order to predict the weather event. These reports are generated by the application.

**6.3.4 Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. This flow can be sequential, branched or concurrent. The activity diagram is suitable for modelling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues or any other system.

The basic notations of an activity diagram are:

**Action states**

Action states represent the non-interruptible actions of objects. They are represented using a rectangle with rounded corners.

**Initial State**

A filled circle followed by an arrow represents the initial action state.

uml_initial.gif

**Final State**

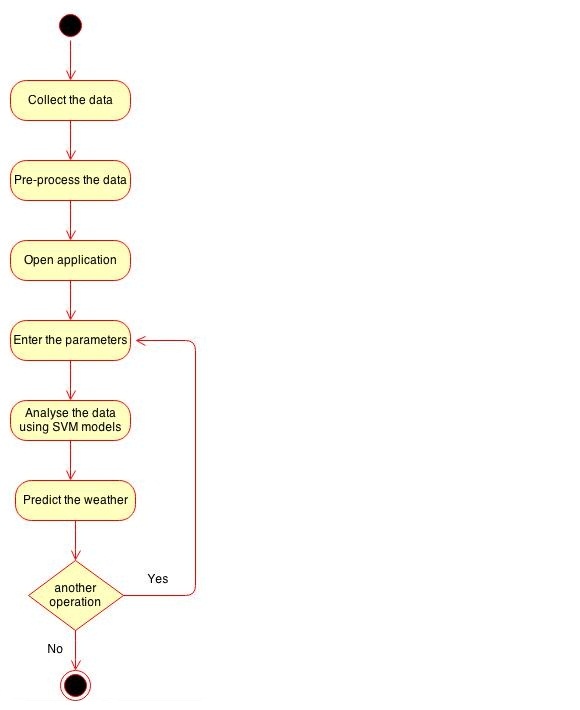
An arrow pointing to a filled circle nested inside another circle represents the final action state.

uml_final.gif

##### Action Flow

Action flow is illustrated using arrows that show the relationships among action states.

arrow.gif



**Description of Activity diagram:**

* Start
* Initially, collect the data.
* The collected data is pre-processed.
* The application is opened.
* Next, enter the parameters in the application.
* Analyse the data using the SVM models created.
* Predict the weather event.
* If there is another operation/event to be predicted, then the control goes back to the fourth step, where again the new parameters are to be entered.
* The remaining procedure is the same.
* Stop