1. **INTRODUCTION**

**1.1 OVERVIEW**

According to the USGS (U.S. Geological Survey), natural disasters and their aftershocks claimed the life of thousands of people. Many of the deaths were caused by landslides which also blocked important routes for emergency response and evacuations. Natural disaster assessment is an important step towards hazard and risk management. Disaster prediction can be a valuable tool for planning first response and emergency evacuations. Unlike the USA, countries like India, Nepal do not have extensive infrastructure to measure and monitor seismic occurrences that could help predict and locate disasters.

Through this project we try to identify high risk zones prone to various disasters based on features that have been studied by research institutions. We use contributing factors as our training set like weather conditions (humidity, rainfall, temperature etc). Several methods like heuristic, semi quantitative, quantitative, and probabilistic and multi criteria decision making process are applied to predict natural disasters. In last few years a change from a heuristic (knowledge based) approach to a data driven approach (statistical approach) was observed with the goal of minimizing subjectivity and increasing reproducibility. Logistic regression model, Discriminant analysis, Multiple Regression Models, Conditional Analysis, Artificial Neural Networks are commonly used methods for mapping. Since in most situations the data is highly unbalanced, a machine learning model is proposed to evaluate the prediction performance with relevant metrics.

**1.2 EXISTING SYSTEM**

Existing Cyclone Prediction Systems works with remote sensing and sensory networks. Remote sensing is the acquisition of information about an object or phenomenon without making physical contact with the object and thus in contrast to on-site observation. Remote sensing is used in numerous fields, including land surveying and most Earth Science disciplines. It also has military, intelligence, commercial, economic, planning, and humanitarian applications. In current usage, the term "remote sensing" generally refers to the use of satellite- or aircraft-based sensor technologies to detect and classify objects on Earth.

The disadvantage with this method is that it can predict the cyclone accurately only about 1-2 days before its occurrence, which might not be enough in relocation at times. Using Machine Learning algorithms with a large amount of data, we can predict the occurrence of cyclone months before the calamity actually occurs.

**1.3 PROPOSED SYSTEM**

In the proposed system consists of two problem statements. The first one being predicting the wind speed across the stations .The time series forecasting algorithms is fed with various training data sets across stations and through accuracy measures we analyze which algorithm works the best for our dataset. This way the time series forecasting algorithms solve the first problem statement.

The second problem statement is the categorization of cyclones into the cyclone types basing on the forecasted maximum wind speed value across various stations.7 categories are available, few of them being super cyclone, severe cyclone, depression etc.

In the proposed method we use both wind speed and pressure attributes unlike other methods where only one of these two were used. This makes our results more effective.

**1.4 ORGANIZATION OF THE REPORT**

**Chapter 2** deals with basic concepts necessary to understand our project like machine learning and disaster management, Chapter **3** deals with the Literature Survey and methodology, Chapter **4** deals with the System Requirements and Specification of the project. This deals with software and hardware requirements of the system for successful project implementation and also the software environment for executing our program and getting the results ,**Chapter 5** deals with the implementation of the proposed system which includes the entire project idea in detail including all the steps preprocessing, training the dataset with algorithms and finding accuracy of the results, **Chapter 6** discusses about the design of the project which is represented in the form of a flow chart, **Chapter 7** deals with the results and recommendation of the project in detail, **Chapter 8** deals with the conclusion and limitations ,**Chapter 9** deals with the future scope and **Chapter 10** deals with Bibliography with all the referred papers and web pages.

**2. BACKGROUND**

**2.1 MACHINE LEARNING**

Machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence (AI). Such tasks involve recognition, diagnosis, planning, robot control, prediction, etc. The changes might be either enhancements to already performing systems or synthesis of new systems. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data over and over, faster and faster is a recent development. Here are a few widely publicized examples of machine learning applications:

i) The self-driving Google car

ii) Online recommendation offers such as those from Amazon and Netflix

iii) Knowing what customers are saying about a brand on Twitter - Machine learning combined with linguistic rule creation.

iv) Fraud detection forms important use of machine learning in our world today.

Resurging interest in machine learning is due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage. All of these things mean it's possible to quickly and automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks. By using algorithms to build models that uncover connections, organizations can make better decisions without human intervention.

Most industries working with large amounts of data have recognized the value of machine learning technology. By gleaning insights from this data, organizations are able to work more efficiently or gain an advantage over competitors.

**2.2 APPLICATIONS OF MACHINE LEARNING**

i) A popular use for machine learning today is pattern recognition because it can recognize many types of images. For instance, the US Postal Service uses machine learning for handwriting recognition.

ii) Banks and other businesses in the financial industry use machine learning technology for two key purposes: to identify important insights in data, and prevent fraud.

iii) Government agencies such as public safety and utilities have a particular need for machine learning since they have multiple sources of data that can be mined for insights. Analyzing sensor data, for example, identifies ways to increase efficiency and save money.

iv) The advent of wearable devices and sensors that can use data to assess a patient's health in real time. The technology can also help medical experts analyze data to identify trends or red flags that may lead to improved diagnoses and treatment.

v) Websites recommending items you might like based on previous purchases are using machine learning to analyze your buying history – and promote other items you'd be interested in. This ability to capture data, analyze it and use it to personalize a shopping experience (or implement a marketing campaign) is the future of retail.

vi) Analyzing minerals in the ground. Predicting refinery sensor failure. Streamlining oil distribution to make it more efficient and cost-effective. The number of machine learning use cases for this industry is vast – and still expanding.

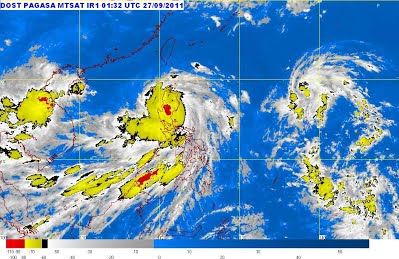
vii) Analyzing data to identify patterns and trends is key to the transportation industry, which relies on making routes more efficient and predicting potential problems to increase profitability.

**2.3 DISASTER (CYCLONE) PREDICTION**

As the name implies, natural disasters are inevitable in our world. Natural disasters include but are not limited to hurricanes, earthquakes, forest fires, tornados, droughts, floods, volcanic eruptions, and tsunamis. They have occurred and will continue to cause problems. Consequently, the idea of “preventing” natural disasters will never be possible. However, the prediction of such disasters is possible with the advancement of technology.

Prediction of natural disasters using technology requires extensive research and funding. Scientists need to analyze past disasters to find patterns in natural phenomena. They study tracking radar systems of seismic waves deep inside the Earth. By comparing past records with live data, specific trends give scientists a warning of future events. Trends are used to predict earthquakes, tsunamis, and volcanic eruptions.

Another way that technology is used in the prediction of natural disasters is with constant surveillance. Using off-shore cameras in hurricane-prone areas ensures that strong winds and waves can be recognised . This also helps to predict tsunamis. Additionally, by monitoring ocean currents, weather patterns can be predicted in advance, warning populated areas under risk of hurricanes and tornados. The surrounding areas can be evacuated, saving many lives. However, these short-term warnings are effective only if relief programs are planned and efficiently carried out.



**Fig 2.1Cyclone Weather Map**

Not only does surveillance help with short term warnings of disasters, but it helps speed up response and recovery teams in their efforts to lessen the effects on society. While this method results in even more short-term warnings than those in use today, short term warnings are considered more reliable. The reliability of any predictions is very important to the process of lessening this global issue. For whichever system or technology used, the results need to have a history of being accurate, or else people do not trust them. When introducing such methods, the public’s trust in the system should be taken into account. In effect, the short term surveillance warnings like the maps shown in fig 2.1 represent the cyclone weather map, although giving people less time to respond, are trusted more because the physical evidence can be given. In contrast, the general public does not have much faith in specific patterns of seismic waves, as few people understand the science and reason behind the predictions.

Our knowledge of how disasters occur, and how they will occur in the future, has never been more sophisticated. We are now able to prophesy impending cataclysms with a specificity that would have been inconceivable just several years ago. Several factors have contributed to this progress: a growing public anxiety about disasters; advances in disciplines as disparate as computer science, fluid mechanics, and neuroscience; and an infusion of funding from governments, universities, and especially corporations, which have figured out that disaster planning saves money in the long run. But the field remains in its infancy. Disaster prediction—like disaster science, disaster economics, disaster-response technology, disaster art, disaster cinema, disaster lit—is a growth industry. All indications suggest a growth curve that will continue to steepen well into the next century.

We are further interested here only in a particular disaster that is Cyclone. Cyclones are most common disasters across the globe. They originate from the coastal regions and are followed by heavy winds and rainfall. Cyclone region can be defined as a low pressure region surrounded by a high pressure region. They follow a trend and pattern .They can be seasonal or random. Wind Speed and Pressure act as contributing factors for cyclones. Further in India cyclone is the most common threat with its vast coastal line.

**2.4 APPLICATIONS OF DISASTER PREDICTION**

Disaster Management cycle involves the following 5 steps and we can incorporate disaster prediction into each and every step (i) prevention where history, corporate memory, and climatology are important; (ii) mitigation that insulates people or infrastructure from hazards; (iii) pre-flood which is the preparation and forecast stage where remote sensing is essential; (iv) response (during the flood) where "actions to be taken is of key importance and weather NOWCASTS (0-3 hour prediction of precipitation) using remote sensing is extremely useful; and (v) recovery (post flood) which is the post-mortem stage where damage assessment, procedures, and numerical weather prediction and hydrological models are validated.

Furthermore by predicting the probability of occurrence of a disaster beforehand life of several thousands of people can be saved. If we are successful in designing an application with low error and high accuracy rate, by seeing the probability of occurrence we could relocate people to safer areas during the disaster prone time. Lives of several thousands of people could be saved by accurate forecasting of the disaster.

Cyclone Prediction can be helpful in issuing warnings along the coasts asking people to vacate the coastal regions. Fisherman can also be warned against going deep into the ocean/sea on a particular day with high predicted probability by an application with low error rate and high accuracy rate.

**3. LITERATURE SURVEY**

Literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

**3.1 LITERATURE SURVEY ON DISASTER PREDICTION**

# Wireless Sensor Networks (WSN) has gained attention as it has been useful in warning about disasters. Predicting natural disasters like hailstorm, fire, rainfall etc. by WSN are infrequent and stochastic. This is an important topic of research. Detection of these disasters should be fast and accurate as they may cause damage and destruction at a large scale. In this paper, comparison of various machine learning techniques such as SVM, regression, decision trees, neural networks etc. has been done for prediction of forest fires. Aditi Kansal et al proposed how regression works best for detection of forest fires with high accuracy by dividing the dataset. Fast detection of forest fires is done in this paper by taking less time as compared to other machine learning techniques.

Flood monitoring is one of the major researches going on in the present scenario with the lot of systems. K. Sakthi Praveen Kumar et al proposed flood prediction based on Machine learning techniques is proposed in our process. The flood prediction system helps in real time monitoring and flood forecasting process. Flood prediction is considered by taking the various parameters such as rainfall, river water level, Capacity, Inflow, Discharge rate, Evaporation loss and river water flow. These values were collected for 15 years from the particular dam. In this work manimuthar dam is considered and the values related to the dam was used to predict the flood in that dam. The input data were initially preprocessed by removing unnecessary information’s like date, duration from the input data. The missing information’s in the data were removed by identifying the gaps in the data. The data is then optimized using Ant colony optimization for the selection of the best data from the input data. The selected data is then used for the classification process. FNN classifier is used for classification process. Finally the performance measures is measured by the calculation of the performance metrics like accuracy, sensitivity and specificity of the classifier

# A large number of extreme floods were closely related to heavy precipitation which lasted for several days or weeks. Long-lead prediction of extreme precipitation, i.e., prediction of 6-15 days ahead of time, is important for understanding the prognostic forecasting potential of many natural disasters, such as floods. Yet, long-lead flood forecasting is a challenging task due to the cascaded uncertainty with prediction errors from measurements to modelling, which makes the current physics-based numerical simulation models extremely complex and inaccurate. Yahui Di et al proposed the modelling work as a machine learning problem and introduce a complementary data mining framework for heavy precipitation prediction.

# In this work, they firstly define the extreme precipitation and non-extreme precipitation clusters and then design the Nearest-Sample Choosing method to handle the imbalanced data sets. They introduce streaming feature selection and subspace learning to extract the most relevant features from high dimensional data. They evaluate the machine learning tools using historical flood data collected in the State of Iowa, the United States and associated hydro meteorological variables from 1948 to 2010.

Carol K. Joseph et al proposed both predicting future disasters, and assessing vulnerability of populations. The goal of such analysis is to prepare governments for emergency response and relief efforts as well as to formulate strategies for future disaster mitigation. In addition, increasing pressures from a growing world population further emphasize the need for governments to seek viable solutions for balancing human needs with environmental constraints. Identifying populations vulnerable to environmental calamities includes analysis of past events to identify patterns which may identify at-risk populations and improve predictions of future events. Past analytical techniques used common statistical methods; however, recent developments in the supervised and unsupervised learning techniques will be applied to natural disaster data extracted from the Emergency Events Database (EM-DAT) for North America, Central America, and the Caribbean.

Unsupervised, descriptive analytics were performed using the density based clustering algorithm DBSCAN to identify notable patterns in the disaster data. A supervised, predictive model was built using an artificial neural network to predict the potential monetary impact of natural calamities based on region, country, and natural disaster type.

# Youssef Safi et al present an application of artificial neural networks to the real-world problem of predicting forest fires. The neural network used for this application is a multilayer perceptron whose architectural parameters, i.e., the number of hidden layers and the number of neurons per layer were heuristically determined. The synaptic weights of this architecture were adjusted using the backpropagation learning algorithm and a large set of real data related to the studied problem. We also present and discuss some examples of illustrating results that show the performance and the usefulness of the resulting neural system.

# In order to reduce the risk of possible damages, governments all around the world are investing into development of Early Warning Systems (EWS) for environmental applications. The most important task of the EWS is identification of the onset of critical situations affecting environment and population, early enough to inform the authorities and general public. Alexander L. Pyayt et al describe an approach for monitoring of flood protections systems based on machine learning methods. Inartificial Intelligence (AI) component has been developed for detection of abnormal dike behaviour. The AI module has been integrated into a EWS platform of the Urban Flood project (EU Seventh Framework Programme) and validated on real-time measurements from the sensors installed in a dike.

# Washington Okori et al examined the application of Machine Learning techniques for famine prediction. Early detection of famine reduces vulnerability of the society at risk. The dataset used in the study was collected between 2004 to 2005 across households in the different regions of Uganda. Dataset from the northern region was found to be most suitable to training datasets of other regions. Classification performance of four methods as Support Vector Machine, K- Nearest Neighbours, Naïve Bayes and Decision tree in prediction of famine were evaluated. Support Vector Machine and K- Nearest Neighbours performed better than the rest of the methods however Support Vector Machine produced the best ROC which can be used by policy makers to determine the cut-off for determining famine prone households. It is recommended in this study that satellite data could be used in combination to show the relationship in prediction of food security as this may increase the specificity of those households at risk.

# 3.2 LITERATURE SURVEY ON MACHINE LEARNING AND ALGORITHMS

**3.2.1 Methodology of Machine Learning**

*3.2.1.1 Identify the needs and set goals*

The first stage is to identify what the machine should be doing. We are simply identifying the need of the machine. When doing that, the first thing to consider is how accurate the system should be. It is best to target for low accuracy at first. Some projects require a high level of accuracy. For example, a surgery robot that is going to cut and stitch people up needs to have very high precision. Some other projects don’t require that much accuracy, like a mobile app that identifies a person’s name from various camera images. So set the accuracy your project demands accordingly.

After you find out the needs, you should set goals according to these needs. The goals of the project should be very specific which will be implemented one by one in the next stage.

*3.2.1.2 Design an end-to-end system*

The second stage is to build an end to end system as soon as possible. This will be a prototype. Keep it in mind that there is no need to over engineer this system. A simple system which can actually produce a usable output for the task you want to solve will be okay in this stage. It doesn’t matter if the output is good or bad. One easy way to build a viable system as a prototype is to copy the state of the art method of an application or project that has already been done and published in the same field. Selecting a system that actually works will reduce the chances of failure. If such an application is not available to be copied, then you need to determine which standard algorithm you should opt to start with.

*3.2.1.3 Data driven adaptation*

This step is data driven refinement and it is the final stage of this methodology. In this step, you need to analyze the system to find out which aspects are working properly and which are not. Use the data you get from analysis to improve the poorly working parts/aspects. Keep in mind that over-engineering and making the system complex will make it hard to manage. Keep things only as complex as it needs to be. If some parts are working as they are supposed to, let them be. If some are not working as you expected it to work, then identify the cause and rectify the issue. Keep in mind that this is the final stage of this methodology. This is where you make your system perfect.

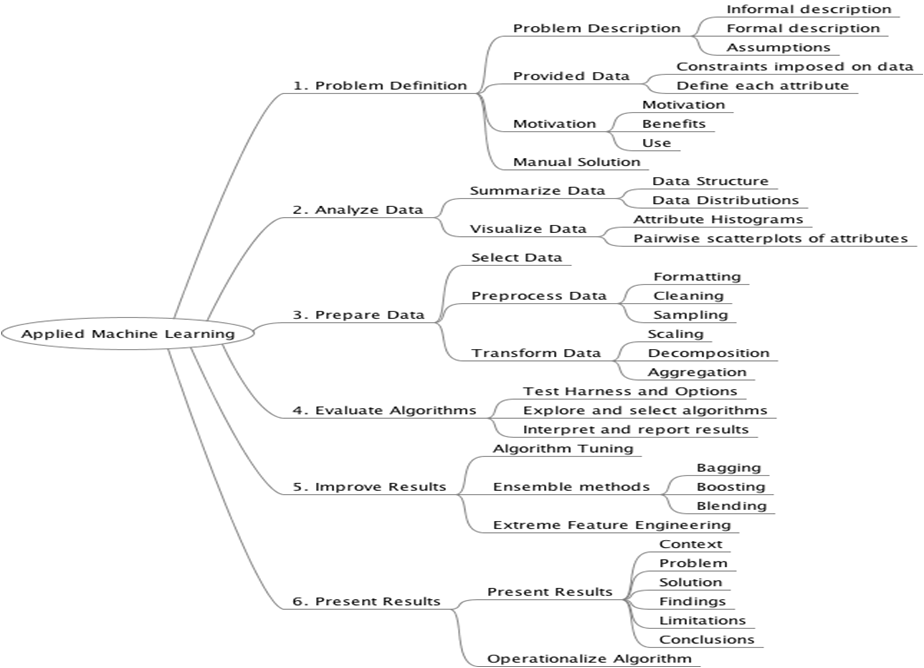
The most important metrics are the training error and test error. Your system should be doing well on the training set and test set. If it’s not good enough in the training set, then it is under fitting. If the test set is not good, then you are over fitting.

*3.2.1.4 Solving High Training errors*

If you are having training errors, the first thing to do is to see if the data has some problems. If the data has defects, then the algorithm won’t work as it is supposed to. Simply check the data and see if there is something wrong with the data that could result in your algorithm giving wrong outputs. After that, inspect the software for bugs. Use popular and trusted software for your project. Do not try to make software of your own if you don’t know what you are doing. If the data and software are without defect, then modify the learning rate and other settings to optimize your system.

*3.2.1.5 Solving high test errors*

If high test errors are present, you can try adding dataset augmentation. This will create multiple copies of your training examples. If regularization strategies like dropout is being used, then you can add dropout up the ladder to reduce test errors. Collecting more data can also solve the problem. After that, inspect the software for bugs. Use popular and trusted software for your project. Do not try to make software of your own if you don’t know what you are doing. If the data and software are without defect, then modify the learning rate and other settings to optimize your system.



**Fig 3.1 Detailed machine learning process**

**3.2.2 Literature survey for algorithms**

Time series forecasting is one of a most widely undertaken research area in statistics. There is a vast literature on the various linear time series models, among which the most popular traditional time series method used for various forecasting tasks is the autoregressive integrated moving average (ARIMA) model. The traditional approach for time series modeling assumes that there is a linear underlying relationship among the past and the future values of a time series. Linear models are very easy to explain and implement, but their major disadvantage is that they may be totally inappropriate if the underlying process is nonlinear. This is the major difference from a nonlinear NN model for which there is no a priori assumptions on the relationship between the variables and the model form is defined by the given data. Xiaowen Zhao a et al proposed a RBF NN model. There are two key features that make NN model a valuable and attractive tool for time series forecasting. First, it represents the data-driven modeling technique with no a priori assumptions about the process under the study. Second, a NN model is a universal functional approximator that it can approximate any continuous function to any desired degree of accuracy.

Recent research activities in forecasting with artificial neural networks (ANNs) suggest that ANNs can be a promising alternative to the traditional linear methods. ARIMA models and ANNs are often compared with mixed conclusions in terms of the superiority in forecasting performance. G. Peter Zhang et al proposed a hybrid methodology that combines both ARIMA and ANN models is proposed to take advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modeling. Experimental results with real datasets indicate that the combined model can be an effective way to improve forecasting accuracy achieved by either of the models used separately.

**4. SYSTEM REQUIREMENT SPECIFICATION**

A system requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements.

**4.1 FUNCTIONAL REQUIREMENTS**

**4.1.1 Training set**

The entire data set is binned into stations. Preprocessing is performed and all the missing values and outliers are removed from the data set. We apply time series algorithms on each of the stations and train the data. The algorithm learns the more as more number of training sets are passed.

**4.1.2 Testing set**

20 wind speed values and corresponding pressure values from each station are fed and next 5 wind speed values are forecasted as output. Using these 5 forecasted wind speed values we try to categorize into cyclone categories (super cyclone, severe cyclone, cyclone etc)

**4.2 NON-FUNCTIONAL REQUIREMENTS**

**4.2.1 Performance Requirements**

* **Usability**: This application allows users to operate with little or no learning. The results can be easily inferred and understood by even laymen.
* **Portability**: The proposed system is portable to work under Java environment imported on windows OS.

**4.3 HARDWARE REQUIREMENTS**

* SYSTEM: Windows 10, 64- bit, i3 Intel Dual Core.
* RAM: 2 GB.
* DISK SPACE: 2-3 GB.

**4.4 SOFTWARE REQUIREMENTS**

* OPERATING SYSTEM: [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows).
* TOOL USED: Eclipse.
* LANGUAGES USED: Python and Java.

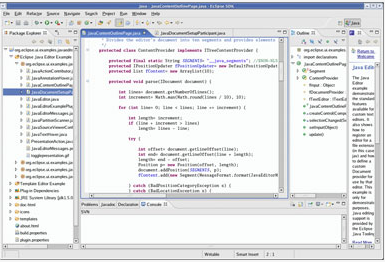
*4.4.1 Eclipse IDE*

Eclipse is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), and is the most widely used Java IDE. It contains a base [workspace](https://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Eclipse is written mostly in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and its primary use is for developing Java applications, but it may also be used to develop applications in other [programming languages](https://en.wikipedia.org/wiki/Programming_language). Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on [Equinox](https://en.wikipedia.org/wiki/Equinox_(OSGi)), an implementation of the [OSGi](https://en.wikipedia.org/wiki/OSGi) core framework specification.

In addition to allowing the Eclipse Platform to be extended using other [programming languages](https://en.wikipedia.org/wiki/Programming_language), such as [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), the plug-in framework allows the Eclipse Platform to work with typesetting languages like [LaTeX](https://en.wikipedia.org/wiki/LaTeX) and networking applications such as [telnet](https://en.wikipedia.org/wiki/Telnet) and [database management systems](https://en.wikipedia.org/wiki/Database_management_system). The plug-in architecture supports writing any desired extension to the environment, such as for [configuration management](https://en.wikipedia.org/wiki/Configuration_management). Java and [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) support is provided in the Eclipse [SDK](https://en.wikipedia.org/wiki/Software_development_kit), with support for other [version control systems](https://en.wikipedia.org/wiki/Version_control_system) provided by third-party plug-ins.

The Eclipse SDK includes the Eclipse Java development tools (JDT), offering an IDE with a built-in Java [incremental compiler](https://en.wikipedia.org/wiki/Incremental_compiler) and a full model of the Java source files. This allows for advanced [refactoring](https://en.wikipedia.org/wiki/Refactor) techniques and code analysis. The IDE also makes use of a *workspace*, in this case a set of [metadata](https://en.wikipedia.org/wiki/Metadata) over a flat filespace allowing external file modifications as long as the corresponding workspace *resource* is refreshed afterward.

Eclipse implements the [graphical control elements](https://en.wikipedia.org/wiki/Graphical_control_element) of the Java toolkit called [Standard Widget Toolkit](https://en.wikipedia.org/wiki/Standard_Widget_Toolkit) (SWT), whereas most Java applications use the Java standard [Abstract Window Toolkit](https://en.wikipedia.org/wiki/Abstract_Window_Toolkit) (AWT) or [Swing](https://en.wikipedia.org/wiki/Swing_(Java)). Eclipse's [user interface](https://en.wikipedia.org/wiki/User_interface) also uses an intermediate [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) layer called [JFace](https://en.wikipedia.org/wiki/JFace), which simplifies the construction of applications based on SWT.



**Fig 4.1 Eclipse IDE**

Fig 4.1 shows the Eclipse IDE and its various components like Project Explorer, Workspace, Console, outline etc. It is a user-friendly IDE that also includes suggestions. It stores projects and classes in a hierarchical format.

**5. IMPLEMENTATION**

**5.1 GENERAL IDEA**

In this project, we take a huge dataset with cyclone Id, Date, Wind Speed, Pressure and other attributes as the input. We then preprocess it to remove the unnecessary attributes which are of no use to us. After preprocessing the entire dataset is divided into stations using binning. We divide this dataset into training and test data. We train our training dataset using classifiers and algorithms and see how well it works on the test data.

The proposed project is divided into two parts. First part is the wind speed forecasting which predicts the wind speed from the pressure values as there exists a linear relationship between them. The second part is the categorization of cyclones into the cyclone type from the wind speed values across various stations. The overall system can be summarized into the following steps:

**5.1.1 Predicting Wind Speed using Pressure across various stations**

The preprocessed data set is binned into several stations based on Station ID.

1. We take wind speed as dependent variable and pressure as independent variable.
2. We take 20 wind speed and corresponding values across various stations and then forecast the next 5 wind speed values.
3. We apply RMSE, correlation and scatter index accuracy measures to analyze which algorithm works the best among all the time series forecasting algorithms.
4. To further improve the results we normalize the data using standard data mining normalization techniques min-max and z-score.

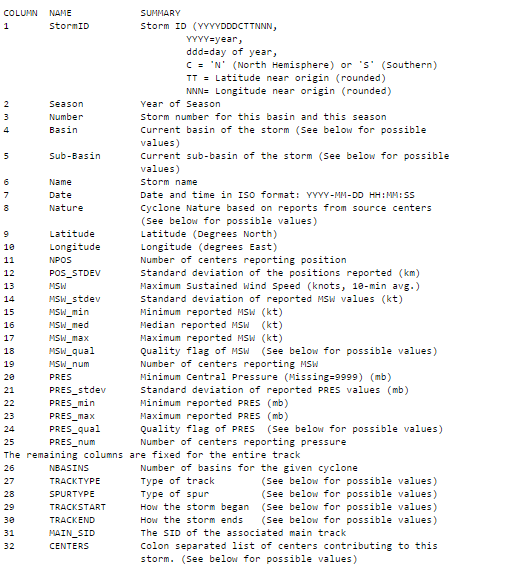
**5.1.2 Categorization of cyclones**

1. Across each station we try to categorize the cyclones.
2. We then categorize the forecasted wind speeds into one of these 7 categories based on the maximum wind speed.
3. Few of the seven categories are Depression, Deep Depression, Cyclone Storm, Severe Cyclonic Storm, Very Severe Cyclonic Storm, Super Cyclonic Storm etc.

**5.2 DETAILED DESCRIPTION**

**i) Module 1(Predicting Wind Speed)**

The first module of the proposed system is about wind speed forecasting. The dataset we are working on is downloaded from national climatic data centre(NCDC).Data set has around 200 attributes with several thousand rows across several stations over the years from late 1900 to 2016.Out of all these attributes very few are useful to us. We perform preprocessing to remove outliers, unwanted attributes and missing values from the entire dataset. Pressure, Station ID, Wind Speed, Date and Year are the only useful attributes shown in fig 5.1.



**Fig 5.1 Dataset attributes screenshot**

We bin the entire dataset into smaller sets using the Station I.e. then train the data in the stations using our time series forecast algorithms. We apply all the four algorithms and train the data. We take 20 wind speed and corresponding pressure values as input and forecast the next 5 wind speed values. We use accuracy measures like root mean square error, correlation and scatter index values to evaluate our forecasted results. Among all the three accuracy measures we rely more on RMSE .Initially RMSE values were found to be very high ,to decrease the values we have opted normalization techniques. We have used Z-Score and Min Max techniques and the RMSE values have decreased significantly across all stations promising relatively accurate forecasts.

**ii) Module 2(Categorization of Cyclones)**

After obtaining the forecasted wind speed values, we find the maximum wind speed value among all the forecasted values and categorize this wind speed value into a cyclone category**.**

**Cyclones are generally classified into one of these 7 categories**

i) Depression category: max wind speed >31 and max wind speed<50.

ii) Deep Depression category: max wind speed >51 and max wind speed<62.

iii) Cyclonic Storm: max wind speed >63 and max wind speed<88.

iv) Severe Cyclonic Storm: max wind speed >89 and max wind speed<117.

v) Very Severe Cyclonic Storm: max wind speed >118 and max wind speed<165.

vi) Extremely Severe Cyclonic Storm: max wind speed >166 and max wind speed<220.

vii) Super Cyclonic Storm: max wind speed >221.

The following are the various stages that are followed in the process of cyclone prediction. Each of these stages play an important role in terms of bringing an optimal result.

**5.2.1 Preprocessing**

Preprocessing is the phase where all the unnecessary information from dataset is removed. It is known that preprocessing can reduce 40% of the input size, thus reducing the burden of extra processing.

The training set and the testing set first underwent the preprocessing stage. This stage consists of the following phases

*5.2.1.1 Removing Unnecessary Attributes*

Our dataset consists of over 200 attributes. Out of all these attributes only few are required for our purpose like Station ID, Wind Speed, Year, Pressure Id and Date. Rest of the attributes like Name, Basins, Track type, Track start, Tracked, Main SID, Centers do not play a major role .Using the dataset without removing them will lead to increased execution time.

*5.2.1.2. Filling Missing Values and Correcting Outliers*

Missing values found in the fields Minimum Wind Speed and Pressure are replaced by -999 .Usually the weather data missing values are replaced with -999 instead of -1.Data out of range are considered outliers and replaced by -999. Here, the outliers considered are MSW>=170, MSW\_STDEV>=30, MCP<870, MCP\_STDEV>=50**.** These value references for outliers are given in the website on whose dataset we are working.

*5.2.1.3 Normalization*

It's simply a case of getting all your data on the same scale: if the scales for different features are wildly different, this can have an effect on your ability to learn. Ensuring standardized feature values implicitly weights all features equally in their representation.

*5.2.1.3.1 Min-Max Normalization*

Min-max normalisation is often known as feature scaling where the values of a numeric range of a feature of data, i.e. a property, are reduced to a scale between 0 and 1. Therefore, in order to calculate *z*, i.e. the normalised value of a member of the set of observed values of *x*, we must employ the following formula.

z =

where min and max are the minimum and maximum values in X, where X is the set of observed values of x.

It can be easily seen that when x=min, then y=0, and When x=max, then y=1.  
This means, the minimum value in X is mapped to 0 and the maximum value in X is mapped to 1. So,the entire range of values of X from min to max is mapped to the range 0 to 1.

*5.2.1.3.2 Z-Score Normalization*

A z-score is the number of standard deviations from the mean a data point is**.** But more technically it’s a measure of how many standard deviations below or above the population mean a raw score is. A z-score is also known as a standard score and it can be placed on a normal distribution curve. Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve). In order to use a z-score, you need to know the mean μ and also the population standard deviation σ.

Z-scores are a way to compare results from a test to a “normal” population. Results from tests or surveys have thousands of possible results and units. However, those results can often seem meaningless. For example, knowing that someone’s weight is 150 pounds might be good information, but if you want to compare it to the “average” person’s weight, looking at a vast table of data can be overwhelming (especially if some weights are recorded in kilograms). A z-score can tell you wherethat person’s weight is compared to the average population’s mean weight.

z=

X is the observation,µ is the mean and is the standard deviation

**5.2.2 Training**

We use time series forecasting algorithms. A time series model can predict trends based only on the original dataset that is used to create the model. You can also add new data to the model when you make a prediction and automatically incorporate the new data in the trend analysis. Time Series Forecasting Techniques are useful when historical data exists for forecast variable and the data exhibits a pattern**.**

Time Series Forecasting models-:

i)Naive forecast.

ii)Period Moving Average.

iii)Linear regression.

iv)Polynomial 4th degree

v)ARIMA

*5.2.2.1 Naive Forecast*

Naive forecast is the simplest time series forecasting model. It is often used as a bench mark to compare with other models. It assumes that tomorrow will be like today .It ignores any other older historic data. Without adjusting or establishing causal factors, using the last periods actual to estimate this period's forecast by this unremarkable technique. A naive forecasting model is a special case of the moving average forecasting model where the number of periods used for smoothing is 1.

*5.2.2.2 Weighted Moving Average*

A moving average forecast model is based on an artificially constructed time series in which the value for a given time period is replaced by the mean of that value and the values for some number of preceding and succeeding time periods. As you may have guessed from the description, this model is best suited to time-series data; i.e. data that changes over time. For example, many charts of individual stocks on the stock market show 20, 50, 100 or 200 day moving averages as a way to show trends.

Since the forecast value for any given period is an average of the previous periods, then the forecast will always appear to "lag" behind either increases or decreases in the observed (dependent) values. For example, if a data series has a noticeable upward trend then a moving average forecast will generally provide an underestimate of the values of the dependent variable.

The moving average method has an advantage over other forecasting models in that it does smooth out peaks and troughs (or valleys) in a set of observations. However, it also has several disadvantages. In particular this model does not produce an actual equation. Therefore, it is not all that useful as a medium-long range forecasting tool. It can only reliably be used to forecast one or two periods into the future. The moving average model is a special case of the more general weighted moving average. In the simple moving average, all weights are equal.

*5.2.2.3 Regression*

Implements a single variable linear regression model using the variable named in the constructor as the independent variable. The coefficients of the regression - the intercept and the slope - as well as the accuracy indicators are determined from the data set passed to init. Once initialized, this model can be applied to another data set using the forecast method to forecast values of the dependent variable based on values of the dependent variable (the one named in the constructor).

A single variable linear regression model essentially attempts to put a straight line through the data points. For the more mathematically inclined, this line is defined by its gradient or slope, and the point at which it intercepts the x-axis (i.e. where the independent variable has, perhaps only theoretically, a value of zero). Mathematically, assuming the independent variable is x and the dependent variable is y, then this line can be represented as

y = intercept + slope \* x

*5.2.2.4 4th Degree Polynomial Regression*

Implements a single variable polynomial regression model using the variable named in the constructor as the independent variable. The coefficients of the regression as well as the accuracy indicators are determined from the data set passed to init. Once initialized, this model can be applied to another data set using the forecast method to forecast values of the dependent variable based on values of the dependent variable (the one named in the constructor).

A single variable polynomial regression model essentially attempts to put a polynomial line - a curve if you prefer - through the data points. Mathematically, assuming the independent variable is x and the dependent variable is y, then this line can be represented as

y = a0 + a1\*x + a2\*+ a3\*+a4\*

You can specify the order of the polynomial fit (the value of m in the above equation) in the constructor.

*5.2.2.5 ARIMA*

ARIMA stands for Auto-Regressive Integrated Moving Averages. The ARIMA forecasting for a stationary time series is nothing but a linear (like a linear regression) equation. The predictors depend on the parameters (p,d,q) of the ARIMA model:

**i)**. **Number of AR (Auto-Regressive) terms (p)**: AR terms are just lags of dependent variable. For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5).

**ii)**. **Number of MA (Moving Average) terms (q)**: MA terms are lagged forecast errors in prediction equation. For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.

**iii) Number of Differences (d):** These are the number of non seasonal differences, i.e. in this case we took the first order difference. So either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.

An importance concern here is how to determine the value of ‘p’ and ‘q’. We use two plots to determine these numbers. Lets discuss them first.

**i) Autocorrelation Function (ACF)**: It is a measure of the correlation between the TS with a lagged version of itself. For instance at lag 5, ACF would compare series at time instant ‘t1’…’t2’ with series at instant ‘t1-5’…’t2-5’ (t1-5 and t2 being end points).

**ii) Partial Autocorrelation Function (PACF**): This measures the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. Eg at lag 5, it will check the correlation but remove the effects already explained by lags 1 to 4.

**5.2.3 Accuracy Measures**

Accuracy Measures are used to evaluate the accuracy of the predicted or forecasted data. Accuracy measures widely used for structured data are RMSE, Correlation and Scatter Index.

*5.2.3.1 RMSE*

The Root-Mean-Square Deviation (RMSD) or Root-Mean-Square Error (RMSE) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed. The RMSD represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable .

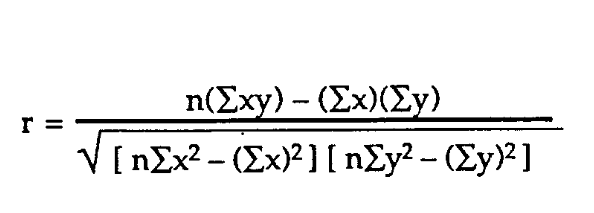
RMSD =

Here n is the number of observations, x1 the actual value and x2 is the predicted value.

*5.2.3.2 Correlation*

Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases.

A correlation coefficient is a statistical measure of the degree to which changes to the value of one variable predict change to the value of another. When the fluctuation of one variable reliably predicts a similar fluctuation in another variable, there’s often a tendency to think that means that the change in one causes the change in the other. However, correlation does not imply causation. There may be, for example, an unknown factor that influences both variables similarly.

****

Here x is the actual value and n is the total number of observations

*5.2.3.3 Scatter Index*

Scatter index is calculated by dividing RMSD with mean of the observations at each grid point and multiplying it with 100. It presents the percentage of RMS difference with respect to mean observation or it gives the percentage of expected error for the parameter. A scatter plot can be used either when one continuous variable that is under the control of the experimenter and the other depends on it or when both continuous variables are independent. If a parameter exists that is systematically incremented and/or decremented by the other, it is called the control parameter or independent variable and is customarily plotted along the horizontal axis. It presents the percentage of RMS difference with respect to mean observation or it gives the percentage of expected error for the parameter.

**5.2.4 Algorithm Overview**

*5.2.4.1 Create dataset*

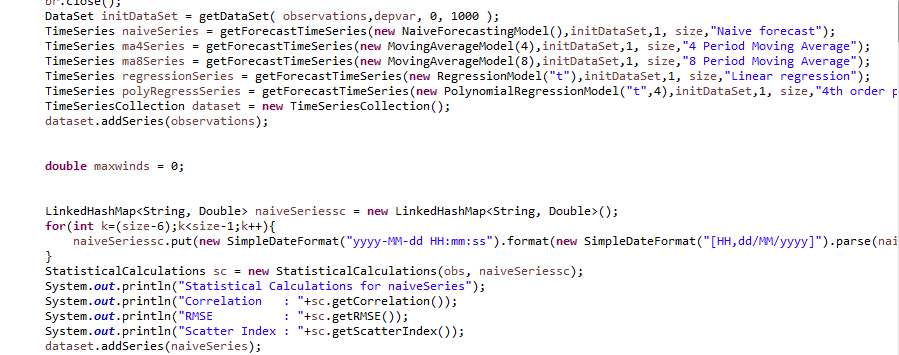
In this function, we are considering the wind speed and pressure values from the dataset (.csv file) which is provided in the parameters. We are formatting the date values into the required format and performing the pre-processing – normalization of data, neglecting the outliers(negative values).

*5.2.4.2. Fitting a Model & Performing Forecast*

In this method, we are fitting the models – naïve, moving averages, regression and poly regression with the observations from the pre-processed data. We are also forecasting the next few values of wind speed using every model. TimeSeries objects are created to store the forecasted values given by each model. We are calling other functions that calculate the accuracy measures – RMSE, Correlation, Scatter Index. We use the functions getRMSE( ), getCorrelation( ), getScatterIndex( ) for this. We have also incorporated the code required for categorization.

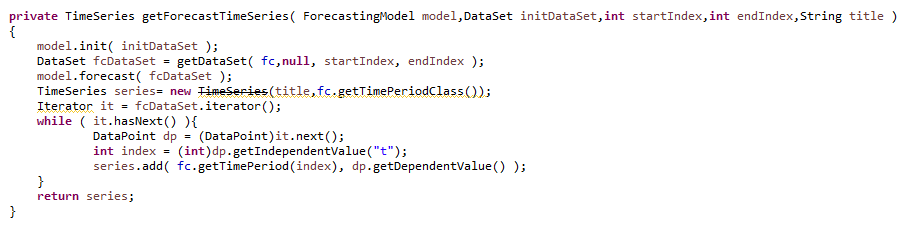
*5.2.4.3. getForecastTimeSeries*( )

We take the model and the preprocessed dataset as parameters and using model.forecast( ) , we perform the forecasts as shown in Fig 5.3. The argument given to this function is a DataSet object. The forecasted results are returned in the form of a TimeSeries object.



**Fig 5.2 Forecasting for each model and calculation of Accuracy measures.**

Fig 5.2 shows the TimeSeries object storing the forecasts from each model and also includes the code for calculating the accuracy measures – RMSE, Scatter Index and Correlation. This is done for each model mentioned.

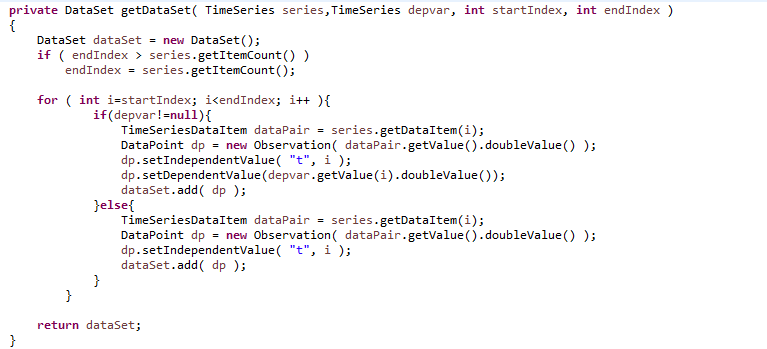


**Fig 5.3 Code snippet of a function getForecastTimeSeries**

Fig 5.3 shows a function called gerForecastTimeSeries that takes the model and dataset as parameters and performs the required training of the model with data.

*5.2.4.4. getDataset*( )

Takes TimeSeries object as argument. A DataPoint object is used to create relation between pressure and wind speed. Pressure is taken as independent variable and Wind Speed is taken as dependant variable as shown in Fig 5.4.



**Fig 5.4 Code to integrate the relationship between independent and dependant variable**

*5.2.4.5. getObservationStandardDeviation*( )

Calculates standard deviation of observations taken into consideration by using another function getVariance( ).

*5.2.4.6. getObservationmean*( )

Calculates mean of observations taken in account.

*5.2.4.7. getForecastStandardDeviation*( )

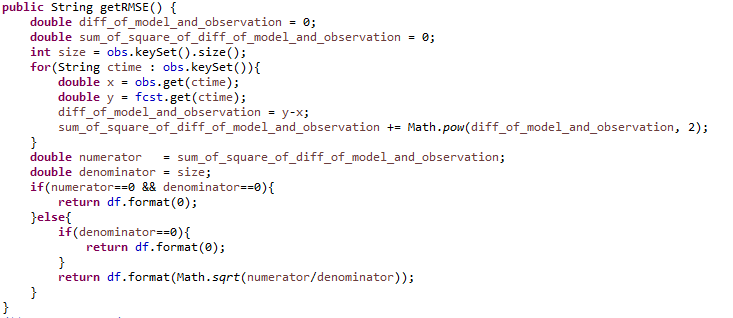
Calculates standard deviation of Forecasted values using the math formula.

*5.2.4.8. getForecastMean*( )

Calculates mean of forecasted *values*.

*5.2.4.9. getRMSE*( )

Using observed and forecasted values, we calculate RMSE values using functions from Math library as shown in Fig 5.5.



**Fig 5.5 Code snippet of a function to calculate RMSE**

*5.2.4.10. getScatterIndex*( )

Using observed and forecasted values, we calculate Scatter Index values using functions from Math library.

*5.2.4.11. getBias*( )

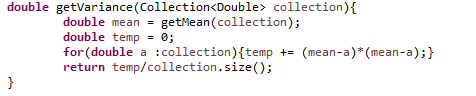
Bias is calculated by subtracting average of observed values from forecasted values.

*5.2.4.12. getMean*( )

Using a Collection object that represents the observations, average is calculated.

*5.2.4.13. getVariance*( )

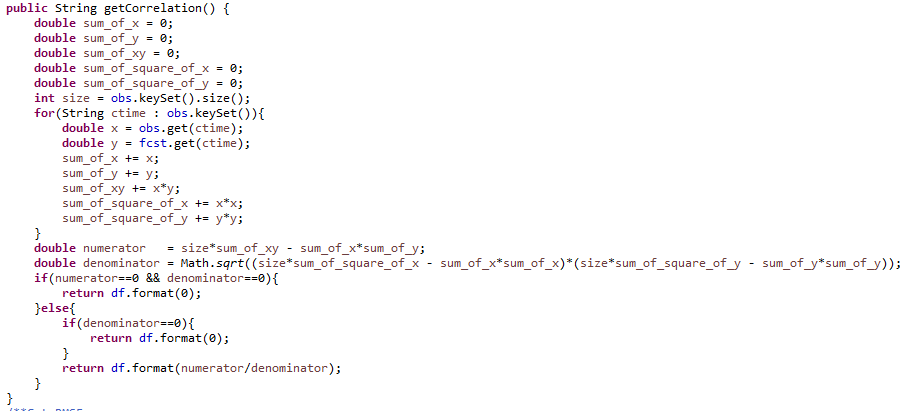
Using Collection object that represents observation and using the Math formula for variance, variance is calculated as shown in Fig 5.6.



**Fig 5.6 Code snippet of a function to calculate Variance**

*5.2.4.14. getCorrelation*( )

This function calculates sum of all predicted values, sum of observed values, sum of product of these two, sum of squares of each of them and substitutes in the mathematical formula of correlation as show in Fig 5.7.

**

**Fig 5.7 Code snippet of a function to calculate Correlation**

**6. DESIGN**

Project design is an early phase of the project where a project’s key features, structure, criteria for success, and major deliverables are all planned out. The point is to develop one or more designs which can be used to achieve the desired project goals.

**6.1 MODULE 1:** Predicting wind speed across stations taking pressure as independent variable.

**Fig 6.1 Flow chart for module 1**

**Module 1:Predicting Wind Speed using Pressure across various stations**

Flowchart 6.1 describes the process of predicting wind speed .

The preprocessed data set is binned into several stations based on Station ID.

1. We take wind speed as dependent variable and pressure as independent variable.
2. We take 20 wind speed and corresponding values across various stations and then forecast the next 5 wind speed values.
3. We apply RMSE, correlation and scatter index accuracy measures to analyze which algorithm works the best among all the time series forecasting algorithms.
4. To further improve the results we normalize the data using standard data mining normalization techniques min-max and z-score.

**Module 2: Categorization of cyclones**

Flowchart 6.2 describes the process of categorization of cyclone into one of the seven categories.

1. Across each station we try to categorize the cyclones.
2. We take maximum wind speed across each of the station and categorize it.
3. We then categorize the forecasted wind speeds into one of these 7 categories based on the maximum wind speed.
4. One of these Depression, Deep Depression, Cyclone Storm, Severe Cyclonic Storm, Very Severe Cyclonic Storm, Super Cyclonic Storm etc.

**6.2 MODULE 2:** Categorization of cyclones

**Fig 6.2 flow chart for module 2**

Categorize it into one of 7 types of cyclone

No Cyclone

no

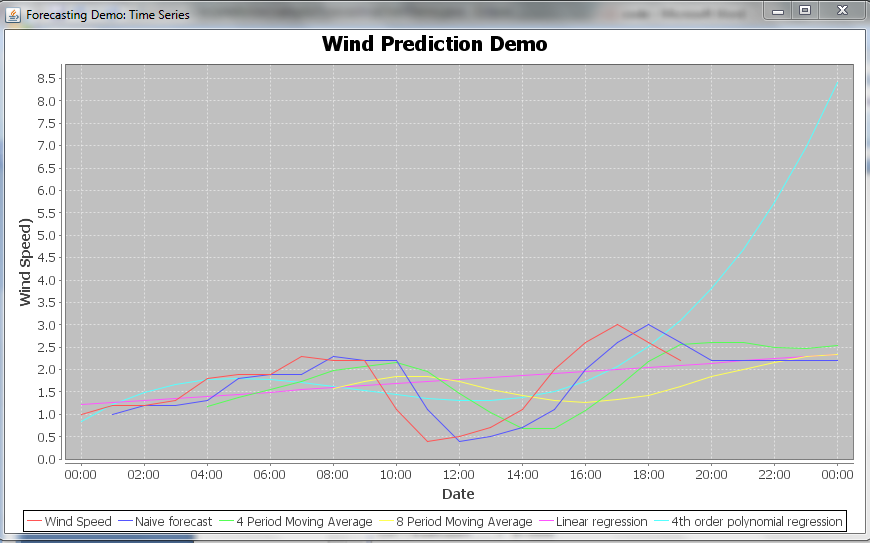
yes

yes

Wind speed belongs to a range

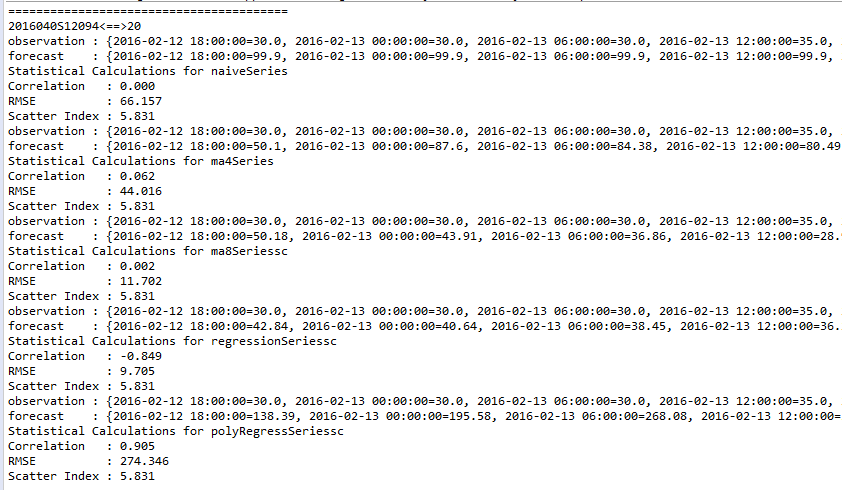
1. **RESULTS AND RECOMMENDATIONS**

**7.1 RESULTS BEFORE NORMALIZATION**

****

**Fig 7.1: Chart denoting trends of various forecasting algorithms**

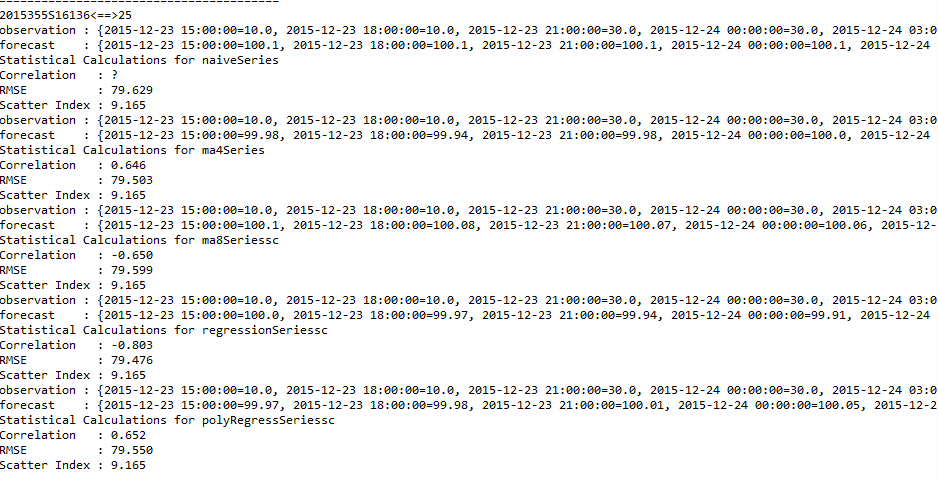
Fig 7.1 shows a graph depicting the actual wind speed values and predicted values from all the time series models we worked on – naïve, moving averages, line regression, and polynomial regression. We plot this graph showing the predicted values by all the different algorithms and from the above graph we can make out that Naive predicts the closest to the actual values. This graph is generated using JFree Charts in Java.

****

**Fig 7.2 RMSE, Correlation and Scatter Index metrics for**

**station with Id 2016040S12094**

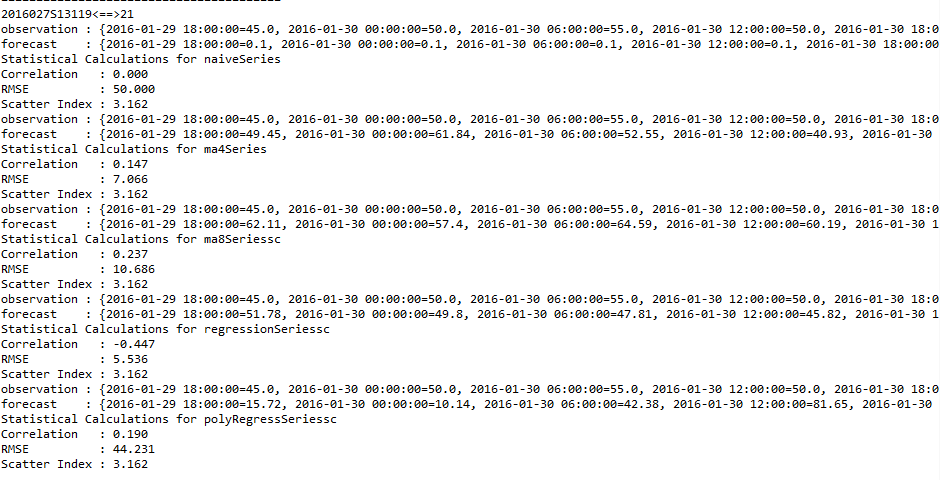
Fig 7.2 illustrates the results in station with Id 2016040S12094 where in the number of observations considered are 20. Basing on these 20 observations the 5 models – naïve series, ma4, ma8, regression series, and poly regression series forecast the next 5 values of wind speed. Various accuracy measures like RMSE, Correlation and Scatter Index are calculated for each model to get a better comparison. Most accurate predictions are found when the correlation values are nearer to +1 which shows that the values are increasing linearly and low RMSE value shows less error for prediction. In the above station lowest RMSE value is 9.78 (regression time series) and highest is 274.346(poly regression time series).Correlation is positive for all the algorithms except regression. Poly regression time series shows has correlation nearest to 1.

****

**Fig 7.3 RMSE, Correlation and Scatter Index metrics for**

**station with Id 2015355S16136**

Fig 7.3 illustrates the results in station with Id 2015355S16136 where in the number of observations considered are 25. Basing on these 25 observations the 5 models – naïve series, ma4, ma8, regression series, and poly regression series forecast the next 5 values of wind speed. Various accuracy measures like RMSE, Correlation and Scatter Index are calculated for each model to get a better comparison. Most accurate predictions are found when the correlation values are nearer to +1 which shows that the values are increasing linearly and low RMSE value shows less error for prediction. In the above stations RMSE value is about 79 for all the algorithms. Correlation is positive for all the algorithms except regression and moving average 8 series .Poly regression time series shows has correlation nearest to 1. Naives shows ? which is a dummy value, this can be observed in very rare cases while working with time series algorithm.



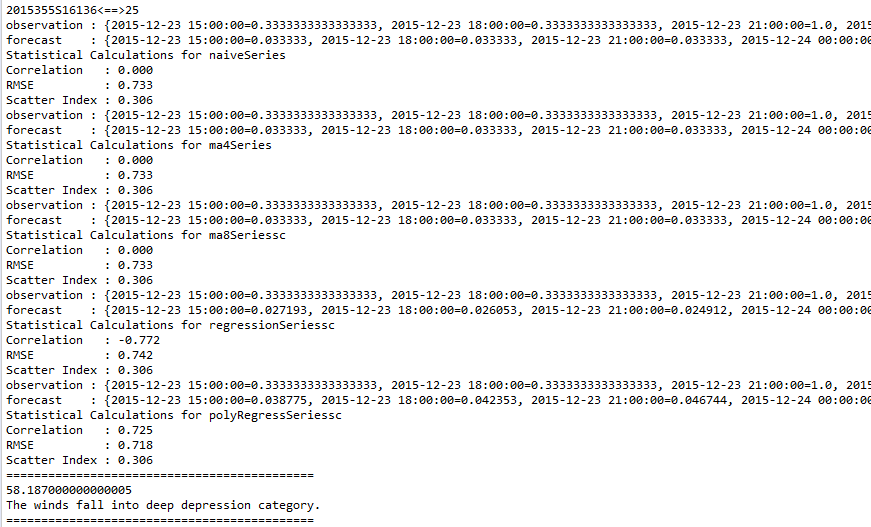
**Fig 7.4 RMSE, Correlation and Scatter Index metrics for**

**station with Id 2016027S13119**

Fig 7.4 illustrates the results in station with Id 2016027S13119 where in the number of observations considered are 21. Basing on these 21 observations the 5 models – naïve series, ma4, ma8, regression series, poly regression series forecast the next 5 values of wind speed. Various accuracy measures like RMSE, Correlation and Scatter Index are calculated for each model to get a better comparison. m8 series has correlation closest to one and regression has negative correlation .RMSE values are ranging between 5.53 and 50.Naive has the highest RMSE as it acts like a bench mark to compare with other time series models.

**7.2 RESULTS AFTER NORMALIZATION**

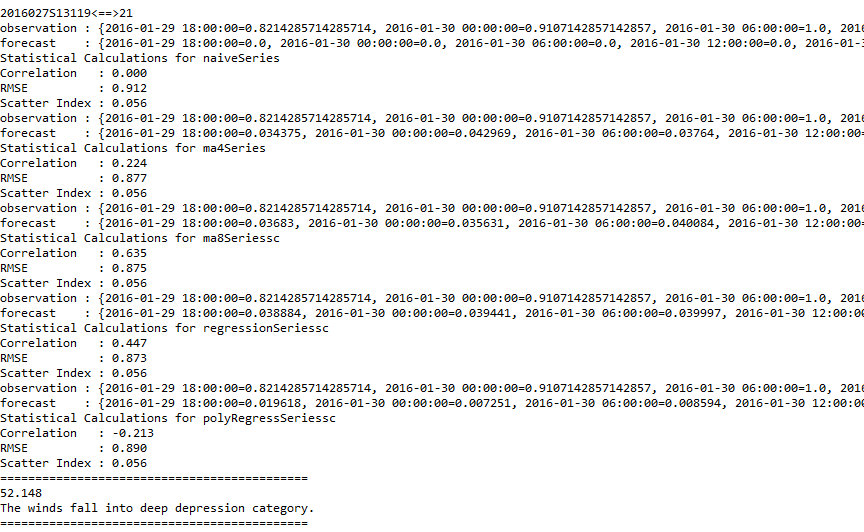
Fig 7.2, 7.3 and 7.4 clearly show high RMSE values and correlation deviations. In order to overcome this we have done normalization on the data. Normalization reduces the deviations between the data and smoothens the overall values. After normalization RMSE values have significantly decreased and correlation values also show a positive non deviating trend overall.



**Fig 7.5 improved values of metrics after min-max normalization for**

**station with Id 2015355S16136**

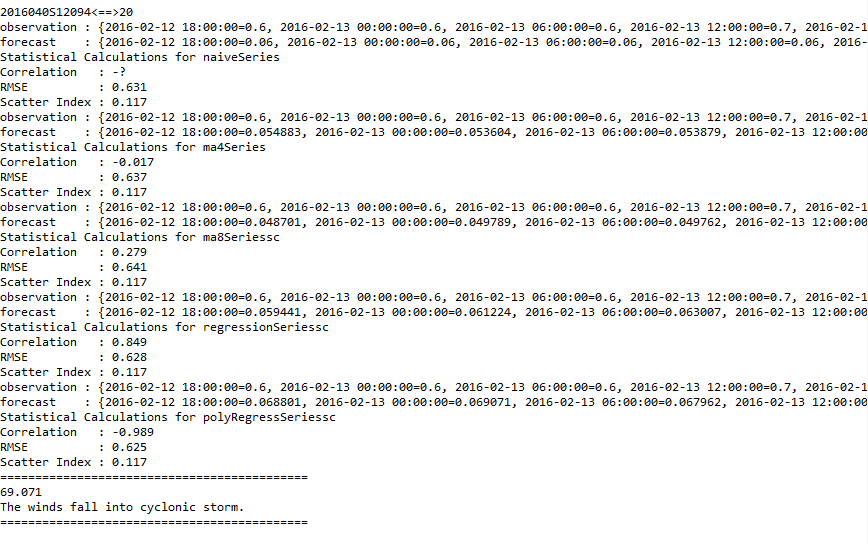
Fig 7.5 illustrates the results in station with Id 2015355S16136 after using the pre-processing technique called min-max normalization technique. Better results are obtained in terms of improved accuracy values for every model. RMSE has decreased significantly. RMSE values which earlier ranged between 274 to 9 now has reduced to mere 0.7 approximately overall. Correlation also shows a positive trend and is closer to +1 showing a linear positive relationship. The above fig 7.5 shows the results of min -max normalization which normalizes the entire set of data between 0-1 reducing the discrepancies.



**Fig 7.6 improved values of metrics after min-max normalization for**

**station with Id 2016027S13119**

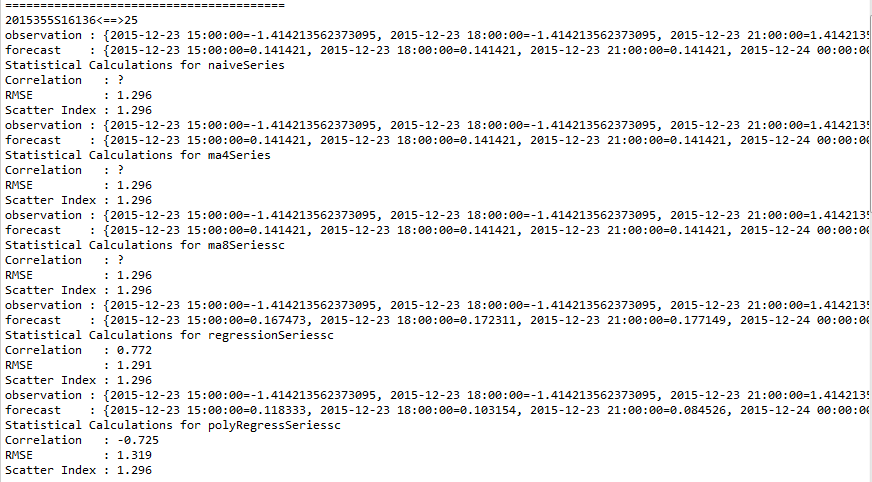
Fig 7.6 illustrates the results in station with Id 2016027S13119 after using the pre-processing technique called min-max normalization technique. Better results are obtained in terms of improved accuracy values for every model. The above fig 7.6 shows the results of min -max normalization which normalizes the entire set of data between 0-1 reducing the discrepancies. RMSE values have decreased significantly from 79 to around 0.89.Correlation also shows a positive trend with close to 0.63.



**Fig 7.7 improved values of metrics after min-max normalization for**

**station with Id 2016040S12094**

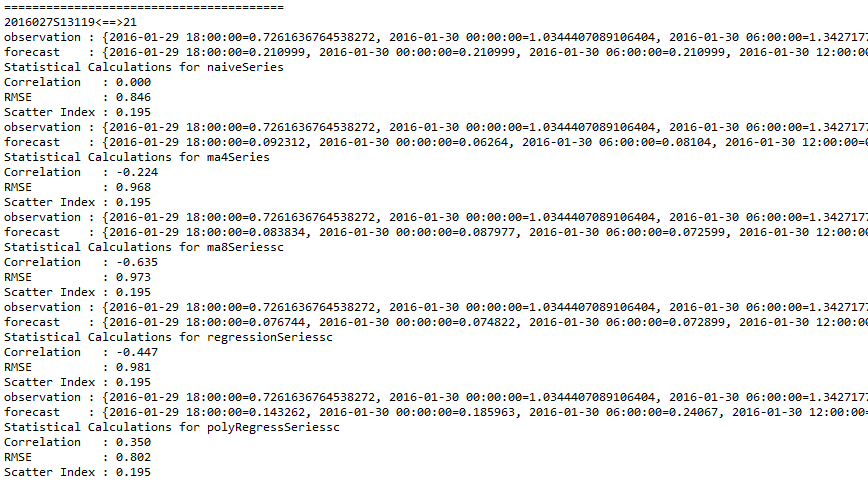
Fig 7.7 illustrates the results in station with Id 2016040S12094 after using the pre-processing technique called min-max normalization technique. Better results are obtained in terms of improved accuracy values for every model. The above fig 7.7 fig shows the results of min -max normalization which normalizes the entire set of data between 0-1 reducing the discrepancies. RMSE values have decreased significantly from 79 to around 0.63.Correlation also shows a positive trend with close to 0.84.



**Fig 7.8 Improved values of metrics after z-score normalization for**

**station with Id 2015355S16136**

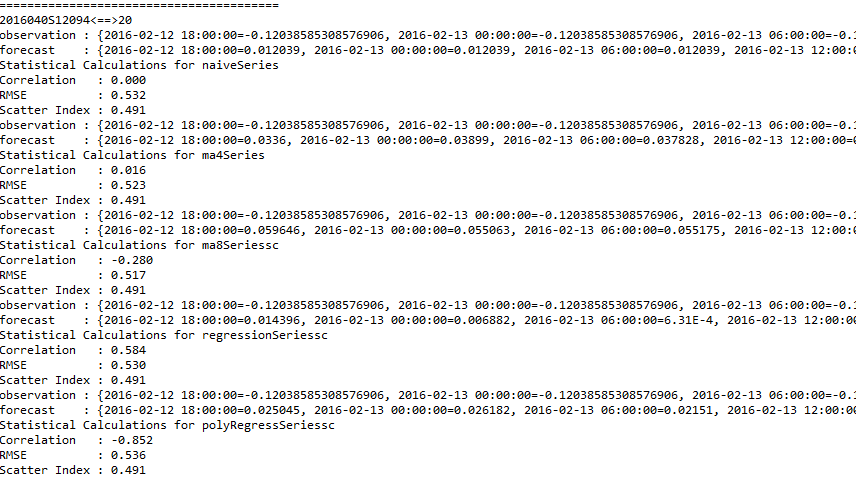
Fig 7.8 illustrates the results in station with Id 2015355S16136 after using the pre-processing technique called z-score normalization technique. Better results are obtained in terms of improved accuracy values for every model. . Z score normalization tries to normalize the data between -7 to +7 on a normal distribution curve. We try to find the best algorithm amongst these two using both correlation and RMSE values. Correlation has a lot of dummy values. Station 2 shows a lot of deviation in values. RMSE value is around 1.3 on an average.



**Fig 7.9 Improved values of metrics after z-score normalization for**

**station with Id 2016027S13119**

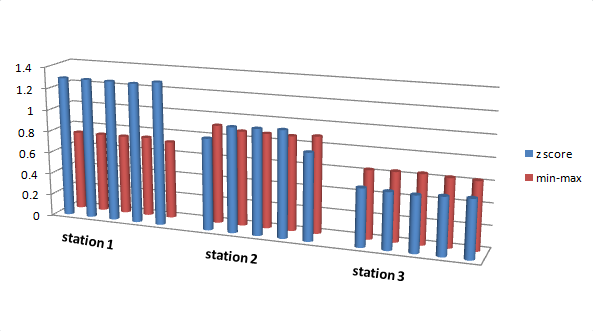
Fig 7.9 illustrates the results in station with Id 2016027S13119 after using the pre-processing technique called z-score normalization technique. Better results are obtained in terms of improved accuracy values for every model. Z score normalization tries to normalize the data between -7 to +7 on a normal distribution curve. We try to find the best algorithm amongst these two using both correlation and RMSE values. RMSE value has decreased from around 70 before normalizing to around 0.89 on an average.



**Fig 7.10 Improved values of metrics after z-score normalization for**

**station with Id 2016040S12094**

Fig 7.10 illustrates the results in station with Id 2016040S12094 after using the pre-processing technique called min-max normalization technique. Better results are obtained in terms of improved accuracy values for every model. Z score normalization tries to normalize the data between -7 to +7 on a normal distribution curve. RMSE values are around 0.5 and correlation shows a positive trend with around 0.58.



**Fig 7.11 Comparison between RMSE values obtained from**

**min-max and z-score normalization**

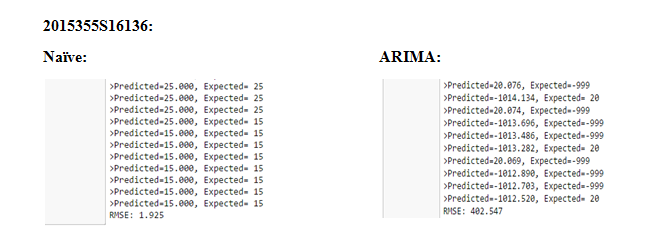
Fig 7.11 shows a comparison of z-score and min-max normalization performance in the 3 stations. It can be concluded that, in majority of cases, min-max shows less RMSE values while in one case, z-score shows less error.

Station 1 has low RMSE when normalized with Min-Max for all the algorithms. Z-score on average for all the algorithms has a RMSE value of 1.2 while Min-Max has around 0.7.The difference is around 0.5.

Station 2 in total has low RMSE when normalized with Min-Max for all the algorithms. Naives and Poly Regression work best with Z score. On average RMSE value for Z-score is around 0.95 and for Min-Max Normalization it is around 0.9.So Min-Max beats Z-Score by 0.05.

Station 3 has low RMSE when normalized with Z-Score for all the algorithms. Min-Max on average for all the algorithms has a RMSE value of 0.6 while Z-Score has around 0.4.The difference is around 0.

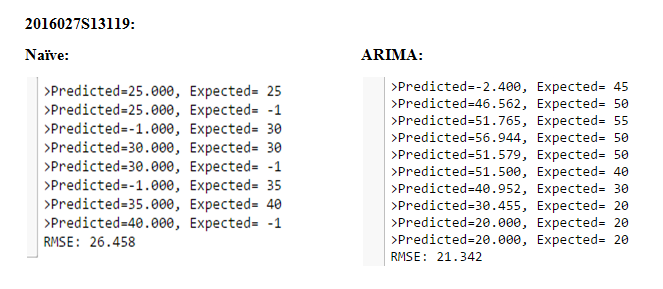
**7.3 ARIMA MODEL**



**Fig 7.12 Persistence model and ARIMA model with RMSE values**

**for station with Id 2015355S16136**

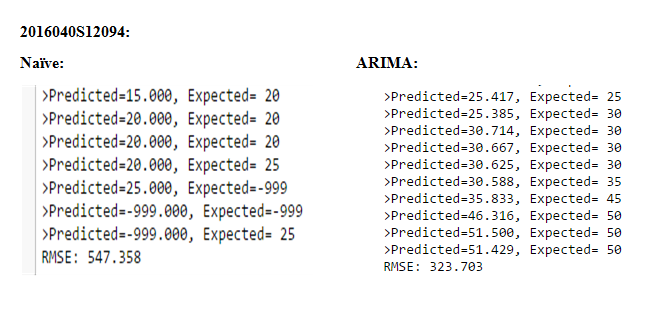
Fig 7.12 describes the results obtained for station with id 2015355S16136 on executing persistence and ARIMA models. RMSE values are calculated to measure the accuracy. Naive is performing better for this model than ARIMA.

****

**Fig 7.13 Persistence model and ARIMA model with RMSE values**

**for station with Id 2016027S1319**

Fig 7.13 describes the results obtained for station with id 2016027S13119 on executing persistence and ARIMA models. ARIMA performs better in this station with RMSE value around 21.



**Fig 7.14 Persistence model and ARIMA model with RMSE values**

**for station with Id 2016040S12094**

Fig 7.14 describes the results obtained for station with id 2016040S12094 on executing persistence and ARIMA models. ARIMA performs better than Naive. These are the predicted RMSE values before normalization. Prediction will significantly increase after normalization.

**7.4 CATEGORIZATION OF CYCLONES**

Categorization is performed on predicted on the maximum predicted wind speed across the station. In all the three stations performing Min-Max and Z-Score normalization, two values are generated across each station and tabulated as in table 7.1. Categorization is performed and wind speed is placed in one of the seven categories.

As a result of data in this report, following recommendations can be made:

Data should be preprocessed properly before fitting the model. This prevents the result to be error prone. Preprocessing like – handling missing values and handling the outliers like the negative wind speed values in this case needs to be done.

Also, for better results and less RMSE values, normalization of data can be performed. This increases the accuracy of results.

**Table 7.1 Category of cyclone in each station for both normalizations**

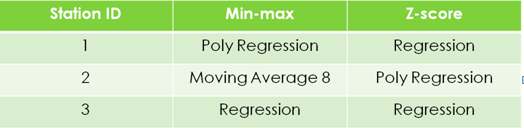
|  |  |  |
| --- | --- | --- |
| **Station Id** | **Min-Max** | **Z-score** |
| 2015355S16136 | deep depression category | extremely severe cyclonic storm |
| 2016027S13119 | deep depression category | super cyclonic storm |
| 2016040S12094 | cyclonic storm | deep depression category |

**8. CONCLUSION AND FUTURE SCOPE**

Firstly to conclude which model works best, we have to consider both RMSE values and correlation values that are calculated. RMSE is the best accuracy measure for comparing performance of time series forecasting models. Correlation gives the deviation between the predicted and observed value and hence proves useful too.

For any model that works best, RMSE values will be less and correlation (i.e. correlation between predicted and observed values) should be near to 1 or equal to 1. Taking these constraints into consideration, we have compared these values for all models in each station.

**Table 8.1 Results of model yielding less RMSE in both cases**

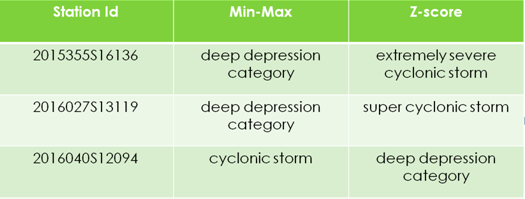


Referring table 8.1, we can observe that Regression model yielded better results i.e. less RMSE and correlation values almost near to1. In majority of cases, Regression seemed to be performing better and hence it can be concluded as the best model. However, Poly Regression model has given good results in two cases and so can be considered as the next best model.

Using these results categorization is performed and the results from categorization are also summarized into a table.

As a final result, at each station the category or the stage of cyclone is predicted. The table 8.2 summarized the results obtained in the 3 stations we considered. Using this information, it would be easier to plan the disaster prevention or mitigation phases.

**Table 8.2 represents the results after categorization**



We would like to further extend the project to other disasters. We could also extend the scale by extending the project to other countries that are more prone to disasters. With the developments like these that take place technologically; we can reduce the effect of disasters to some extent. We have used supervised learning algorithms; other unsupervised algorithms can be tested further to see if they work better. Unsupervised algorithms like k-means clustering, Random Forest etc could be applied and the results could be observed. We could connect our results to any of the IOT device which would give the probabilities of disaster just like weather on our phone. We could also integrate these into a mobile application so as to give updates about probabilities of occurrence to the people easily. We have preprocessed our dataset which can be used by people for other purpose.

**BIBLIOGRAPHY**

# Aditi Kansal, Yashwant Singh, Nagesh Kumar and Vandana Mohindru, ”Detection of Forest Fires Using Machine Learning Technique”, IEEE Journal 2016

1. K. Sakthi Praveen Kumar, Dr.M.Pallikonda Rajasekaran, “Flood Prediction using Optimized Machine Learning Techniques”, IEEE Conference Paper 2016
2. Alexander L. Pyayt, Ilya I. Mokhov, Bernhard Lang, Valeria V. Krzhizhanovskaya and Robert J. Meijer, “Machine Learning Methods for Environmental Monitoring and Flood Protection”, IEEE Journal 2011
3. Youessef Safi et al, “Prediction of Forest Fires using ANN”, IEEE Journal 2013

# Time series forecasting using a hybrid ARIMA and neural network model(G. Peter Zhang 2001)

1. http://robotics.stanford.edu/people/nilsson/MLBOOK.pdf.
2. http://www.sas.com/en\_us/insights/analytics/machine-learning.html.
3. http://media.bemyapp.com/practical-methodology-deploying-machine-learning/.
4. https://data.gov.in/catalog/number-cyclonic-stormssevere-cyclonic-storms-formed-over-north-indian-ocean.
5. https://azure.microsoft.com/en-in/documentation/articles/machine-learning-algorithm-choice/.