Development of AI/ML Based early warning system for vector-borne diseases outbreaks using meteorological and health parameters

A Thesis

submitted to

Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Rishabh Dev
20171196

&   &

Indian Institute of Science Education and Research, Pune
Indian Institute Of Tropical Meteorology, Pune
India Meteorological Department, Pune

April, 2022

Supervisor: Rajib Chattopadhyay, IITM-IMD, Pune
TAC Expert: Pranay Goel, IISER Pune
Co-Supervisor: Bhupendra B Singh, IITM, Pune

Certificate

This is to certify that this dissertation entitled *Development of AI/ML Based early warning system for vector-borne diseases outbreaks using meteorological and health parameters* towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Rishabh Dev at India Meteorological Department (IMD), Pune and Indian Institute of Tropical Meteorology (IITM), Pune under the supervision of Rajib Chattopadhyay, IMD, during the academic year 2017-2022.

Rishabh Dev

Digitally signed by Rajib Chattopadhyay
Date: 2022.03.31 14:34:08 +05'30'

Rajib Chattopadhyay

Committee:

Supervisor: Dr Rajib Chattopadhyay, IMD Pune

TAC Expert: Dr Pranay Goel, IISER Pune

Co-Supervisor: Dr Bhupendra B Singh, IITM Pune

This thesis is dedicated to my grand father

Declaration

I hereby declare that the matter embodied in the report entitled *Development of AI/ML Based early warning system for vector-borne diseases outbreaks using meteorological and health parameters* are the results of the work carried out by me at the India Meteorological Department (IMD), Pune and Indian Institute of Tropical Meteorology (IITM), Pune, under the supervision of Rajib Chattopadhyay and the same has not been submitted elsewhere for any other degree.

Rajib Chattopadhyay
Digitally signed by Rajib Chattopadhyay
Date: 2022.03.31 14:35:47 +05'30'

Rajib Chattopadhyay

Rishabh Dev

Rishabh Dev

Acknowledgments

I would like to acknowledge and give my warmest thanks to my supervisor Prof. Rajib Chattopadhyay who made this work possible. His guidance and advice carried me through all the stages of writing my project. I would also like to thank my co-supervisor, Prof. Bhupendra B Singh and Prof. Bipin Kumar and Prof. Manmeet Singh from IITM Pune for providing me space, opportunity and continuous guidance throughout my project. I would also like to thank my committee member, Prof Pranay Goel, for his brilliant comments and suggestions, thanks to you.

I have a special thanks to my brother in law Mr Deepak Singh who gave me immense support and technical guidance whenever needed. He has guided me through every technicality and approach in data science which led me to get this project. He is my role model.

I would also like to give special thanks to my mother and sister who are pillars of my life. I would also thank my family members for their continuous support and understanding when undertaking my research and writing my project. Your prayer for me was what sustained me this far. I would also like to thank my friends who gave me immense support and some incredible stories to cherish throughout my life in these five years.

In the end, I want to say thanks heartily to IISER Pune for creating this wonderful atmosphere where everyone can learn and grow in their zones. I want to thank all the professors, TAs and support staff of IISER Pune who guided me through every stage of learning and provided me with a beautiful time at IISER Pune.

Abstract

In India, high prevalence of vector borne disease like malaria in certain region is a significant health burden and require extensive community monitoring surveillance and management. Proliferation and spread of all vector borne diseases (including malaria) is known to be tied with several environmental and meteorological factors in addition to human, parasite and vector biology. Modelling and predicting malaria cases is, thus, essential as it can provide an early warning and improve community response in treating the disease. Various time series modelling techniques can be used to model malaria. In this study, we have used Machine learning-based methods like Linear and ridge regression, Self-Organising Maps (SOMs) and LSTM to generate forecast or qualitative district wise outlooks for malaria cases in Bihar. We have used meteorological and health parameters. Our LSTM model for malaria prediction is very flexible and accurate. We have also developed a dashboard for data and model predictions which can later be used as a product for malaria prevention.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vector-borne diseases(VBDs) are illnesses caused by infectious pathogens transmitted into the host body by other living organisms causing multiple diseases in human population [1]. Organisms carrying these pathogens are called vectors. Every year we globally record a over million deaths and more than one billion infection caused by vector-borne diseases like malaria, dengue, schistosomiasis, human African trypanosomiasis, leishmaniasis, Chagas disease, yellow fever, Japanese encephalitis and onchocerciasis. Among all other infectious diseases, VBDs alone contribute 17% cases anually.

1.1 Background

The prevalence of these diseases is determined by the complex evolution of environmental and social factors. Global trends in tourism and trade, random urbanisation, and environmental challenges such as climate change have had a major impact on disease transmission in recent years. Other diseases, such as dengue, chikungunya, and West Nile virus, appear in lands where they were previously unknown. Changes in agricultural practices due to varying temperatures and rainfall may affect the transmission of vector-borne diseases. Weather and climate information can be used to monitor and predict the distribution and long-term conditions of malaria and other climate-sensitive diseases[2]. This study will focus on Malaria.

1.1.1 Malaria Life-cycle

Malaria is caused by parasites of the genus *Plasmodium* and it consists of several species. The complex life cycle of *Plasmodium* takes place in two phases; sexual and asexual, the vector mosquitoes and the vertebrate hosts. In the vectors, mosquitoes, the sexual phase of the parasite's life cycle occurs. The asexual phase of the life cycle occurs in humans, the intermediate host for malaria. Human malaria is transmitted only by female mosquitoes of the genus *Anopheles*. Life cycle of a *Anopheles* has been shown here:

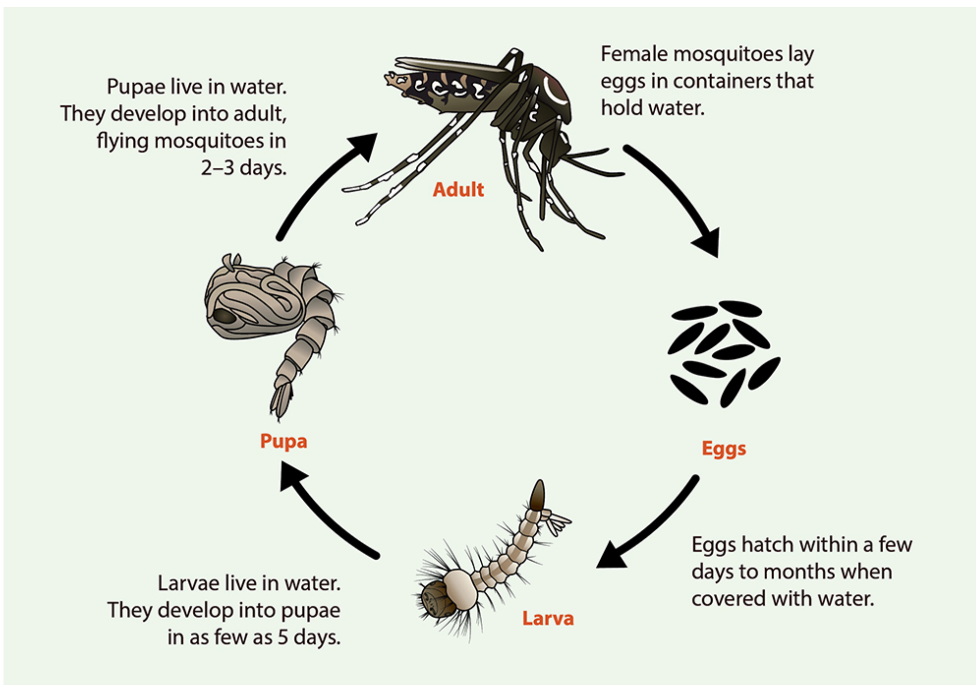


Figure 1.1: Life Cycle of a mosquito. Source: cdc.gov[3]

The malaria transmission dynamics depends on the life-history and population parameter of vector which decides the survival of adult *Anopheles* mosquitoes[4]. Studies have shown decreased larvae survival with the increase of temperature. Higher environmental temperature can also significantly lower the adult vector survival overall. An increase in humidity at a constant temperature in the lab conditions have lowered the total mean eggs laid by female vector[5]. Overall at higher temperatures and lower humidity, it has been seen a significant drop in the vector population.

1.1.2 Malaria as a Global Threat

Malaria is a vector-borne disease. Malaria resulted in 209 million infections and 409 thousand deaths globally in 2019[6]. Most deaths occur among children living in Africa where a child dies every minute from malaria. Malaria mortality rates among children in Africa have been reduced by an estimated 54% since 2000. Since 2015, there is an steep increase in the total malaria cases worldwide[7]. This burden of disease and death resulted in more than 100 global efforts and research to improve the prevention, diagnosis, and treatment of malaria. The global trends in malaria cases and mortality has been shown in Fig. 1.1.

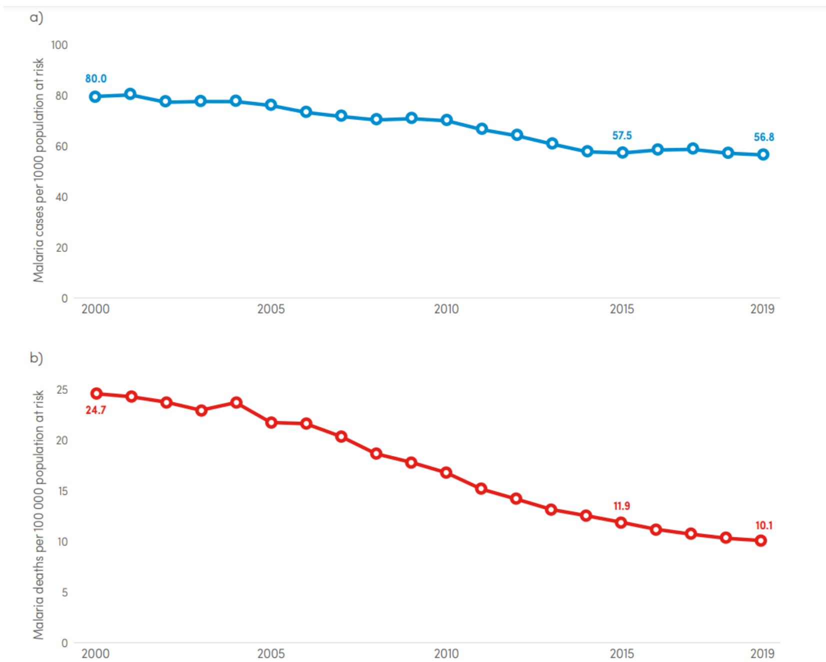


Figure 1.2: a) Global malaria case incidence rate (cases per 1000 population at risk), b) Global mortality rate (deaths per 100 000 population at risk), 2000–2019. Source: WHO estimates[8]

1.2 Malaria in India

In the WHO's Southeast Asian province, of the 1.4 billion people living in 11 countries, 1.2 billion are at risk of contracting malaria, most of whom live in India[**?**]. However, Southeast

Asia has contributed only 2.5 million cases of global malaria burden. In this regard, India alone has contributed 76% of the total cases. The WHO Regional Office in South East Asia estimates 15 million cases and 20,000 deaths. Most of the malaria burden is borne by years of economic productivity.

The National Framework for Malaria Elimination (NFME)[9] serves as a framework to road-map the outline for malaria elimination by 2030. Under this program, every state in India has been divided into zones based on severity of malaria prevalence. Based on this categorisation, economically poor states like Bihar and Odisha fall into category of high severity. Out of these states, Bihar is India's most flood-prone state, with 76% population in the North Bihar living under the recurring threat of flood devastation. Extreme flood increases the risk of malaria contraption by 30%[10]. In a study, it has been reported that malaria causes most deaths in the age group of 14-39 years and for 70+ age group, malaria causes second most number of deaths in Bihar[11].

1.2.1 Malaria Prevention Programs

India has several National Diseases Control/Prevention plans in action. Most of these programs are funded by the government and function in a hierarchy[12]. Some programs include operating networks of primary health care systems at Panchayats and urban health posts or dispensaries in the towns functional under municipal councils. These primary health care systems are the first line of action towards any outbreak, including testing, prevention and reporting active cases to the secondary line of defence located in district hospitals. They are also responsible for spreading awareness among the masses regarding understanding and prevention of the outbreak. The tertiary line of action in the chain is medical colleges and hospitals. Tertiary defences handle severe cases and using the collected data along the tree, they plan accordingly.

1.2.2 Socio-economic Burden of Malaria and Need for an Early Warning System

Malaria has a profound social and economic impact on developing countries and tropical nations like India. There was a survey from independence that included the financial burden of malaria[13]. It has been found that millions of cases are reported each year, and in this case, one third of the workforce is active and productive adults. This results in financial loss to the individual depending on the income. The spread of malaria and economic losses were studied in steel mines in the Sundargarh region of Orissa[14]. It is estimated that 11,04,841 Rupees (US $ 24,282 at current prices) were lost in three mines due to malaria in 1988. In 1994, Shiv Lal and others estimated that if no control measures were in place and malaria was allowed to pass from 1947, there would be Rupees 76,600 million (US $ 1,670 million) for medication, medical advice, hospitalisation, and absenteeism[15]. That is a result of the fact that the cost of regulatory measures is significant, given the disability, mortality, economic loss and industrial inequality that poor people and the country face. There is a growing need for mechanisms that will allow for early forecasting and early warning of detection of cases in unstable transfers so that more control measures can be implemented effectively.

1.3 Meteorological factors and VBDs

In areas where VBD most frequently occurs, suitable physiological conditions must occur for the vector, host, and pathogen survival and reproduction/replication. These physiological conditions are likely to be affected directly or indirectly by meteorological parameters. Gubler et al. list a range of possible mechanisms related to temperature, rainfall and humidity which impact the risk of transmission of VBD[16].

1.3.1 Temperature

The most important factor in development of malarial vector is the survival of larvae. Studies have shown that incubation period of malarial parasites is temperature sensitive. Studies have proposed various thermodynamic malaria development model to demonstrate that temperature fluctuation can substantially alter the incubation period of the parasite, and hence malaria transmission rates[17]. A study in Africa 2000-2001 have shown that temperature changes have greatly contributed to the fluctuations in the malaria risk pattern using a spatio-temporal model and satellite imagery[18]. Thus, temperature fluctuations may lead to change in the rate of vector population, survival, behaviour, susceptibility, incubation period and seasonality of vector activity.

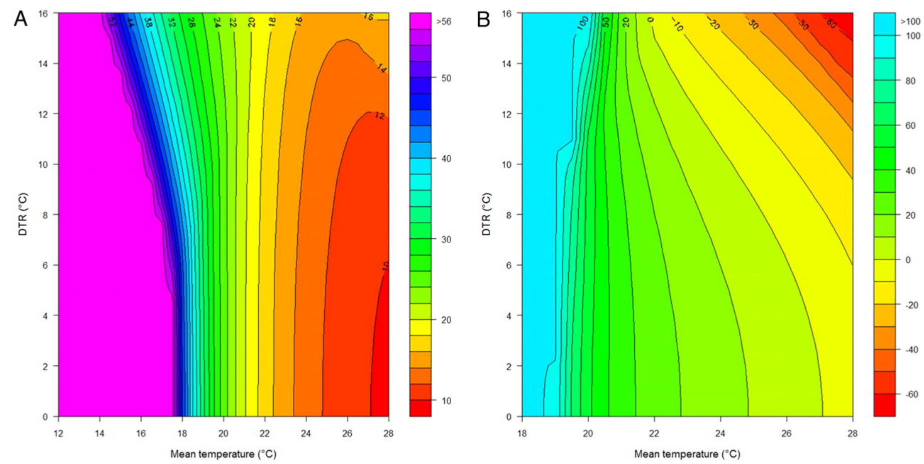


Figure 1.3: Changes in the extrinsic incubation period of malaria parasites and in the basic reproductive number as a consequence of temperature fluctuation. (A) Duration of the extrinsic incubation period (days, right hand bar) of Plasmodium falciparum parasites across a range of mean temperatures (12–28°C) and diurnal temperature ranges (0–16°C). (B) The relative change in R_0 (%, right hand bar) across a range of mean temperatures (18–28°C) and diurnal temperature ranges (0–16°C), comparing R_0. Source:Paaijmaans et. al [17]

Some studies and climate projects have suggested that a window based on the maximum and minimum temperature can give a proxy of malaria prevalence in a region[19][20]. WHO have also recognised these efforts and stated a general temperature window for each mosquito species. As per WHO, Malaria (Plasmodium falciparum) has a maximum temperature(Tmax) window of 33-39 degrees Celsius and 16-19 degrees Celsius for minimum temperature (Tmin). In this study, we will take these temperature windows as threshold temperatures.

Risk	Temperature Window	
	Tmax 33-39 °C	Tmin 16-19 °C
High	✓	✓
Medium	X	✓
Medium	✓	X
Low	X	X

Table 1.1: Temperature Window for Malaria Transmission

If both Tmax and Tmin range are satisfied for a region, we will recognise it as the area of high risk. We call it a moderate risk for either of Tmax and Tmin conditions holds, and a risk-free zone will be an area where no conditions hold. These malaria transmission risks are shown in Table 1.1. A tick represents that the condition has been satisfied in the table, and a cross represents the opposite.

Later in this study, we have used temperature window-based malaria transmission as one of our forecasting methods.

1.3.2 Precipitation and Humidity

Mosquitoes carrying malaria pathogen have breeding grounds in stagnant water or marshy land. Malaria dynamics is closely related to the water cyclicity and it's seasonality. Gubler et al. have suggested that precipitation can also play a significant role. An increased rainfall can increase the water level and provide a broader breeding ground whether as decreased rainfall can make water stagnant in lakes and rivers which again contributes towards an increased malarial activity. Rainfall can also increase vegetation which will allow expansion of vector population. Flooding can also impact the malaria transmission. Heavy rainfall can wash away the breeding grounds but also force a close contact with humans. A greater relative humidity can increase vector activity which in turn increases the malaria transmission.

Later in this study, we have used our data set to perform a power spectral density for malaria, rainfall and humidity, and we have observed that the dominant peaks for variables are nearby, thus confirming a relation. More details are in the next chapter.

1.4 Forecasting Malaria

Malaria studies have shown their association with climatic conditions such as rain patterns, temperature and humidity[21][22]. In many places, the transfer is seasonal, with the highest number of times during and after the rainy season. Accurate forecasts and early warning signs of an increase in the burden of disease can provide the necessary information on the use of targeted malaria control and prevention strategies that make the best use of limited resources. The malaria early warning system (MEWS) is considered a promising tool for preventing / reducing the burden of malaria by calculating the severity of the unexpectedly changing malaria-climate problems. MEWS priorities should include

- case studies (early detection of epidemics)
- preliminary weather-based warnings
- annual weather forecasts.

1.4.1 Climate Change and Malaria

Climate change is now a constant crisis for the whole world. Any prolonged variation in the mean state of the climate can be referred to as climate change[23]. Since 1950, global temperature has constantly risen at an unprecedented rate that is causing global warming, resulting in irregular rain patterns and typical scenarios of drought and famine and other such misfortunes. Few elementary models have suggested that this will cause an increase in malaria cases globally[24]. Because of the complex relationship between malaria and climate change, there will be an increase in the chances of malaria transmission in areas where malaria is common, in areas where the disease is under control, and in areas where there has been no malaria as the climate changes[25]. Increased temperatures, rainfall, and humidity may increase the number of malaria-carrying mosquitoes in the highlands, resulting in an increase in malaria transmission in areas not previously reported. In low-lying areas where malaria is already a problem, warmer temperatures will alter the parasite's growth cycle for mosquitoes to grow faster, increasing the infection and thus contributing to the burden of disease.

1.4.2 Previous approaches

In recent decades, many kinds of statistics predicting the incidence of malaria in recent months have been made. Exponentially weighted moving average models, auto-regressive integrated moving average (ARIMA) with seasonal features, and seasonal combined frequency (SARIMA) models used in historical demographic data to assess their ability to estimate the number of malaria cases a few months in advance[26]. These type of time-series modelling only suitable for a stationary or seasonal process and not ideal for predicting non-seasonal data. Other methods designed to integrate using remote sensing data have been used to achieve the goal. Xgboost and SVM based methods have also been used but they are only effective in higher dimensions[27]. Few Binomial models that only uses statistical significance have been used in Nigeria to serve the purpose[28]. In Kenya, spatial-temporal methods employed Bayesian statistic to predict indicators of malaria transmission, such as entomological import rates[29]. The accuracy of the models varies and is difficult to interpret in the studies given the diversity of forecasting methods used, including the way in which models are evaluated.

1.4.3 Seasonality and Non-Linearity in Malaria Forecasting

In the zoonotic diseases like Malaria, seasonality plays an important role between vector and host. Vectors and their hosts are subject to seasonal variations that are climate related (e.g. temperature) and climate independent (e.g. day-length). Seasonal variations affect abundance and demographic processes of both vectors and hosts. Vectors can be affected by the way in which temperature can change from season to season, with resultant impacts on their development, activity, and disease transmission role. The lifecycle and activity level of the host can be affected as well, affecting how fast infected or immune animals die and how fast uninfected animals are borne, with resultant impacts on the epidemiology of vector-borne zoonoses.

Vector seasonality due to temperature affects development and activity of vector which will affect the transmission. Host demographic processes (reproduction, birth and mortality rates), affected directly by weather and indirectly by resource availability which will drive the VBD epidemiology. The relationship between vector transmission and host contingency is complex and gives a non-linear behaviour. Rogers and Randolph have shown a quadratic relationship between temperature and vector mortality. Hence, in the forecasting models that this study will adapt, will focus on non-linear methods. A linear model hypothesis can't be true. A non-linear model with complex seasonality can be achieved by using the artificial neural networks.

1.4.4 AI/ML based Methods

In the previous approaches, we know that an underlying non-linear function is required to establish a relationship between malaria and other variables. In these methods, we fix our mathematical function and slightly tune parameters to get a good function that can approximate the data. Supervised machine learning is good function approximators given the input data that can map an underlying function without specifying the function beforehand. Function approximation is a technique for estimating an unknown underlying function using historical or available observations from the domain. In function approximation, the exact function is still unknown.

" The Universal Approximation Theorem tells us that Neural Networks has a kind of universality, i.e. no matter what f(x) is, there is a network that can approximately approach the result and do the job! This result holds for any number of inputs and outputs." - Neural Networks and the Universal Approximation Theorem - tds [30]

In this study we aim to use multiple AI/ML based methods to forecast malaria cases. Using multiple methods gives us the flexibility to choose between the best models and use them according to different demographics. The key objectives of this study is as follows:

- Use Bihar as our model state to investigate the malaria cases
- Build models in hierarchy with increasing complexity
- Tune models to get reasonable accuracy on real-time data
- Build a state-of-the art model for forecasting malaria
- Develop an end-user malaria prediction product for non-technical individual

Chapter 2

Methodology

In this project, we have developed three models: rainfall prediction model, Malaria Prediction model and temperature window-based malaria forecasting model. Details will be shared in subsequent sections. Our pipeline will have a few standard pipelines and some unique adaptations. Unique adaptations are marked wherever used, while standard pipelines are stated as a general method.

For Rainfall and Temperature prediction we have used only meteorological parameters (rainfall, Tmax, Tmin, Humidity and Soil Moisture) and health data is used only for malaria prediction.

2.1 Work Station

For all data pre-processing and model building steps, we have used Python 3.8 as our programming language and VS-code by Microsoft as our Integrated development environment (IDE). Data pre-processing work has been done in the Linux environment using Zorin OS which is built on the top of the Ubuntu 20.04 LTS for better file and package management. For model development and deployment, we have used Tensorflow 2.3, open-source software library from the developers of Google Brain team. This model pipeline has been adopted in a hybrid system of windows powershell and Windows subsystem for Linux (WSL) in Windows 11.

2.2 Data Acquisition

Our model primarily focuses on using meteorological and malaria data to forecast future cases. It is established that malaria is sensitive to climate[31]. Rainfall, temperature and humidity are the main factors contributing to driving malaria epidemics.

2.2.1 Health Data

We have got access to district wise health data of Bihar state which is collected from tertiary health centres, namely district hospitals and medical institutes via IMD, Pune. IMD has received data from Bihar govt health department under a national collaboration to develop health climate outlooks. Data has been weekly since 2013 and pre-processed. It has been seen that data varies very drastically between districts and is subject to medical infrastructure allocated for data collection and reporting. Thus, Time-span of available data is 8 years (2013-2020) and Temporal resolution is weekly.

2.2.2 Meteorological data

In the model building, rainfall, maximum temperature (Tmax) , minimum temperature (Tmin), humidity and Soil Moisture is included along with weekly health data. Rainfall data collected from IMD data repository . Data format is NetCDF and 0.25 x 0.25 resolution. We have collected data from 1979 - 2020.

Humidity, Soil Moisture (Amount of water in the vegetation canopy and/or in a thin layer on the soil) and 2m temperature (the temperature of air at 2m above the surface of land, sea or inland waters) data collected from ERA5 Reanalysis data repository. Data is hourly and format is NetCDF. Pre-processing has been applied to convert daily and hourly data into weekly for our region of interest which will be discussed next. Hourly Reanalysis data is only available from 1979, so we have to restrict our rainfall data from 1979. During pre-processing, we will convert 2m temperature data into two new variables, Tmax and Tmin which refers to maximum and minimum temperatures.

So, we are using Rainfall data from IMD repository and Tmax, Tmin, humidity and

	Rain	**Tmax**	**Tmin**	**Humidity**	**Soil Moisture**
Source	IMD Data	ERA5 Data	ERA 5 Data	ERA 5 Data	ERA5 Data
Derived From	Rain	2m Temperature	2m Temperature	2m Temperature	Skin Reservoir Content
Unit	mm	Kelvins	Kelvins	Percentage	m of equivalent water
Horizontal Resolution	0.25 X 0.25	0.25 X 0.25	0.25 X 0.25	0.25 X 0.25	0.25 X 0.25
Time Span	1979 - 2020	1979 - 2020	1979 - 2020	1979 - 2020	1979 - 2020
Temporal Resolution	Daily	Hourly	Hourly	Hourly	Hourly

Table 2.1: Summary of Meteorological Data

Soil Moisture data from ERA5 data repository. Since, Malaria Data (or Health Data), is available only for 2013 - 2020, so we will crop our meteorological data accordingly where malaria data will be used. Now we have 5 climate variables(Rain, Tmax, Tmin, Humidity, Soil Moisture) and 1 variable corresponding to Health. Each variable has a NetCDF file that contain the information about the variable with coordinates.

2.3 Standard Data pre-processing Pipeline

This will be a general pre-processing pipeline for all three models. Main motif of data prepossessing here to transform the NetCDF file of each variable and get a .csv file for each variable where each column should represent the districts and rows should represent the weeks. The spatial information associated with the NetCDF file will be used to get values for each district. For Health data we will get 416 rows corresponding to each weeks from 2013 to 2020 and for rest of data we will get 2192 rows. For development of malaria models later, we will take the bottom 416 rows from meteorological data and whole health data for a chosen district. At the end of pre-processing, we will get a .csv file for each variable whose each column contain data corresponding to each district in Bihar.

2.3.1 Health Data

Health data has already been pre-processed but we have separate files for each district. Each file has columns representing weeks and rows representing weeks ranging from 1-52. So, using pandas library, we have to concatenate the columns over each other and append this single column to a new csv file. looping this over all the districts, we get a single file where rows contain the weeks and columns represent the districts. Now, we have 38 columns and 416 rows.

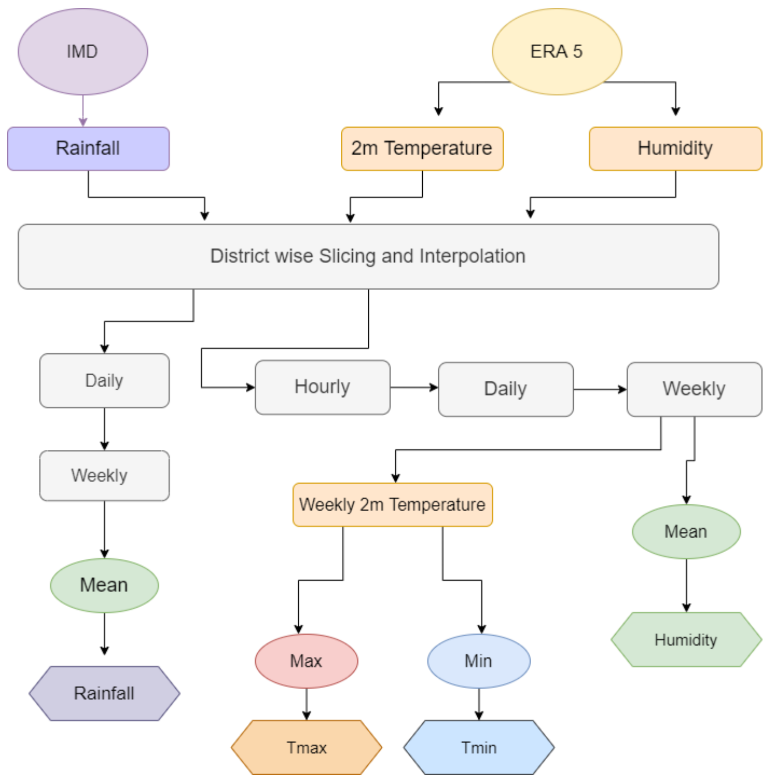


Figure 2.1: Meteorological Data Pre-processing Pipeline

2.3.2 Meteorological Data

For reading the NetCDF and Binary data, we have used the Xarray library. Loading the data in this way gives us a Xarray data-array, which is functionally the same as the Numpy array in python. We have rainfall data of the entire Indian region, so we used shapefiles to mask out the area of interest, i.e., district wise rainfall for Bihar state. A shapefile is a vector data storage format for storing geographic features' location, shape, and attributes. Using Xarray and Rio-xarray library in python, we masked the rainfall data for each district. Since the data resolution is 0.25 x 0.25, we interpolated the data 4 times using the xarray library that laterally expanded the data over more grids by taking mean. After masking and interpolating, we can concatenate the data into a single CSV. Similar pre-processing was applied to Tmax, Tmin and humidity data. After pre-processing, we have a CSV file for each variable. Fig 2.1 shows the data-preprocessing pipeline.

The pre-processing pipeline for Soil Moisture is same as Humidity, so it has been omitted in the above flowchart. A plot (Fig. 2.1) of Rainfall for year 2020 was generated using the processed data and python as a plotting tool.

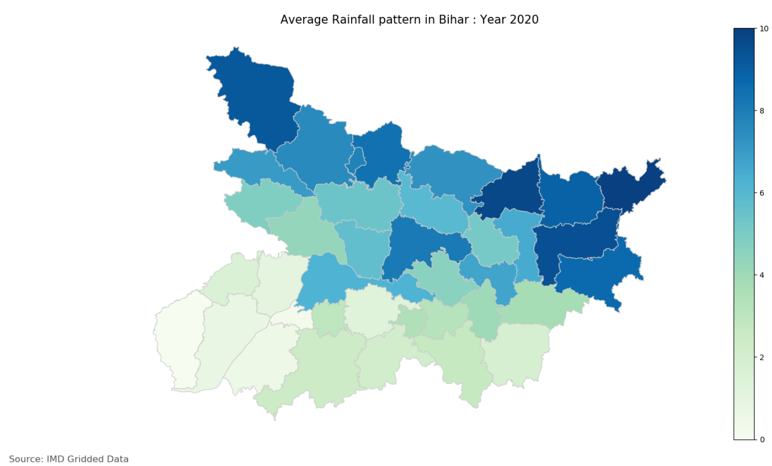


Figure 2.2: District-wise Average Rainfall for year 2020 in Bihar State

2.4 Approaching the model development

Before moving towards actual model building and specific data pre-processings, let us briefly understand how we approach the problem.

We consider our prediction task a time series problem in which we generally have some independent time series which are then modelled to make predictions. Many traditional time-series-based models exist like ARIMA and SARIMA, but we only focus on using Machine learning techniques in this project. We plan on using Linear Regression, SOM Regression and LSTM based neural network models.

We have planned to make prediction models which take input of just past four data points and make predictions on the following four points. To understand this situation better, let us simplify predicting rainfall using past rainfall records only. So we will take the input of the past four-week rainfall value and try to predict the fifth-week rainfall value. This forecast will be a lead time 1-week forecast:

$$Rainfall(t,\ \ t-\tau,\ \ t-2\tau,\ \ t-3\tau) \longrightarrow\ \ Rainfall(t+\tau)$$

Subsequently, if we try to forecast the sixth week, we call it a lead time 2-week forecast:

$$Rainfall(t,\ \ t-\tau,\ \ t-2\tau,\ \ t-3\tau) \longrightarrow\ \ Rainfall(t+2\tau)$$

If we try to capture this harmony, we need to create the input and output data during model training such that we have a lag of 1 data point in the output column for lead time one-week forecast and so on. In order to create this continuous lag and feed it into the model, we have used some libraries in Tensorflow, which will automate the process (details are in the next section).

Next, we will discuss the type of output variable used for the LSTM models. In a regression task, we generally use continuous variables to regress/predict the continuous values, while in a classification task, we predict the classes. We have continuous variables to perform regression tasks. However, we used the upper and lower tercile values to assign class (among 1, 2 and 3) to each continuous variable and performed a classification task. This classification is same as we have seen in Table 1.1. Hence, we have two modes for our rainfall and malaria predictions: series (for regression task) and Categorical (for classification task). For temperature based malaria forecasting, we have used just series task and then used classfication based on table 1.1.

2.5 Data Preparation as per Model requirements

This section onwards will focus on building deployable models for each district. In order to achieve this, we need to prepare data (possibly a pandas data frame) for each district using the CSV files obtained from the general pre-processing pipeline obtained from the above section. After successfully importing the district data, we will create a Time series generator and some model-specific pre-processing.

Pandas Data Frame Generation

First, we will import CSV files for each variable. Each column contains district-specific values recorded in the rows over time. Using pandas, we have concatenated these columns over a single data frame. Pandas is a very convenient library in Python, which is generally very fast and reliably for small data sets like ours, and many model inputs have an excellent pipeline built-in to work hand-in-hand with a pandas data frame. In the end, we will get a pandas data frame that has all the meteorological and health parameters values for the district of interest. For most model performance visualisations, we will be using data for the Patna district.

For Rainfall and Temperature window prediction, we have taken the 41 years of data(Rain, Tmax, Tmin, Humidity and Soil Moisture). Malaria prediction is based on eight years of data (Rain, Tmax, Tmin, Humidity and Malaria). If the output variable is categorical for classification tasks, we have to add another column in the data frame for the same.

Data Splitting

For rainfall and temperature window prediction: we have access to 41 years of data from 1979 to 2020. We have used data from 1979 - 2010 for training the models (both series and categorical mode) and 2011 - 2020 as a testing or validation period.

For malaria prediction: Health data has range of 2013 - 2020 with a total 416 data points. We have used the bottom 416 points of the meteorological data and built the data for splitting. 2013 - 2018 data used for training and 2019 - 2020 used for test or validation

Prediction Leadtime: 1-4 week lead time				
Forecast	Rainfall	Malaria	Rainfall	Malaria
	Training Set		Testing Set	
Linear Regression	1979-2010	2013-18	2011-20	2019-20
SOM Regression	1979-2010	2013-18	2011-20	2019-20
Series LSTM	1979-2010	2013-18	2011-20	2019-20
Categorical LSTM	1979-2010	2013-18	2011-20	2019-20
Fourier LSTM	1979-2010	2013-18	2011-20	2019-20
	Temperature		Temperature	
Temperature Forecast LSTM	1979-2012		2013-20	Remark: Used for Malaria transmission window based forecast as per WHO (Sec 1.3.1)

Table 2.2: Model wise Training and Test Data

data.

We split the data into the training and testing sets using the built-in splitting module called train-test-split in the sklearn library. The splitting is end-to-end, so there is no shuffling of data points in the series. After the splitting, we get four arrays corresponding to independent and target variables for training and testing.

Data Normalisation

In table 2.1, we have seen that all the variables have a different unit, which may cause our ML model to give more weight/significance to larger values during prediction. We will normalise the data by dividing and multiplying the variables by their respective minimum and maximum values to remove this bias. This operation can be done uniformly using the Sklearn module called MinMaxScalar, which is based on the following algorithm.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

	Sampling: Weekly			
Forecast	Rainfall	Malaria	Rainfall	Malaria
	Predictand		Predictor	
Linear Regression	Rainfall	Malaria	Rain, Tmax, Tmin Humidity	Rain, Tmax, Tmin Humidity, Malaria
SOM Regression	Rainfall	Malaria	Rain, Tmax, Tmin Humidity	Rain, Tmax, Tmin Humidity, Malaria
Series LSTM	Rainfall	Malaria	Rain, Tmax, Tmin Humidity, Soil Moisture	Rain, Tmax, Tmin Humidity, Malaria
Categorical LSTM	Rainfall	Malaria	Rain, Tmax, Tmin Humidity, Soil Moisture	Rain, Tmax, Tmin Humidity, Malaria
Fourier LSTM	Rainfall	Malaria	Rain, Tmax, Tmin Humidity, Soil Moisture	Rain, Tmax, Tmin Humidity, Malaria
	Temperature		Temperature	
Temperature Forecast LSTM	Tmax, Tmin		Rain, Tmax, Tmin, Humidity	Remark: Used for Malaria transmission window based forecast as per WHO (Sec 1.3.1)

Table 2.3: Model wise Predictor and Predictands

The scaling or normalisation should be done after the splitting of data because if we try to calculate minimum and maximum on the un-split data, then we already include the real data statistics, which will ruin the idea of dividing the data into training and 'unseen' testing sets. So we need to perform this normalisation individually on each training and testing.

Creating lagged data for training

Tensorflow Keras has a built-in module called TimeSeriesGenerator, which generates the lagged data in batches. It takes three arguments: the number of variables in the predictand set, the number of prior data used for training (4 for our case) and the batch size (we defined it as 32). This module will generate batches of lagged data by only one point using the defined number of prior data and predictand set. In order to create lag or lead time greater than one week, we will shift the whole data frame before putting it into the TimeSeriesGenerator.

For example, if we want to create a lead time 2-week data, first we will obtain the pandas data frame and then shift the predictor set by 1. We will inject this data frame into our Keras module, which will use a four-week window to create the lag of one more point. Eventually, we will get batches of lagged data four weeks prior, corresponding to a lead time two-week data frame used for training.

2.5.1 Power Spectral Density based pre-processing of Malaria Prediction

This pre-processing is only done for Malaria prediction. So this is the only model-specific pre-processing that we have used. Here we will discuss why power spectral density-based pre-processing is needed and how we performed and applied it in the model.

2.5.2 Spectral Analysis

Every time series has a component of periodicity. Periodicity reveals how certain things occur at times. Sometimes the periodicity is very simple that we can easily observe just by looking at it, but it is sometimes difficult to say anything. We use spectral analysis to analyse the complex patterns of periodicity, which requires us to switch from the time domain to the frequency domain.

According to the frequency-domain perspective, a white noise process may be considered the sum of an unlimited number of cycles of different frequencies, each with the same weight. The Fourier theory backs this claim, stating that any continuous function may be written as the sum of an unlimited number of sines and cosines.

Here we will use spectral analysis to denoise the malaria data. We will find the underlying dominant frequencies, and using those frequencies, we will recreate the data, so the minor frequencies or noise gets removed. So our first goal is to find the dominant frequencies. We will use a fast Fourier transform (FFT) in the NumPy library in Python.

$$\begin{bmatrix} | \\ F \\ | \end{bmatrix} \longrightarrow fft \begin{bmatrix} | \\ \hat{F} \\ | \end{bmatrix}$$

We have our data F, which is a vector and we are going to run through this FFT command, and we are going to get this vector $\hat{F}$, complex-valued Fourier coefficients, that tell us how much magnitude and phase of the sine and cosine components of increasing frequency we would have to add up to get this data set. These $\hat{F}$s are complex values with a magnitude and a phase. The magnitude tells us how important that particular frequency is, and the phase tells us if it is more cosine or sine is in the mixture.

Once we have this Fourier transform now, we will compute the power spectral density(PSD), which is just $\hat{F}$ times the conjugate of $\hat{F}$ divided by N. Consider this as multiplication of a complex number λ and it's conjugate, which will give the magnitude of λ squared.

$$\begin{aligned} \lambda &= a + ib \\ \hat{\lambda} &= a - ib \\ \lambda\hat{\lambda} &= |\lambda^2| = a^2 + b^2 \end{aligned} \tag{2.1}$$

Using the above logic, we are computing the magnitude of each Fourier coefficients squared and we are going to get a vector of powers at each frequency, and we can convince ourselves that if this is a function of time, then these each frequency has units of per second these will be in units of Hertz and if we take F hat times its conjugate this will have kind of units of power so this is the power spectral density that we are computing.

Application of Spectral Analysis in Malaria Data

We applied the same pipeline we discussed to transform the data. We take the malaria data for a particular district, pass it through a fast Fourier transform, and multiply the output by its conjugate to get the power spectral density. Below (Fig 2.3) is the plot of the power spectral density of malaria data for the Patna district. The black curve is the actual data, and the cyan curve is the plot of dominant frequencies.

After that, we used the frequencies with the PSD above 100(choice is random, based on visual analysis) and obtained the new signal. This signal has only a few frequencies and is noise-free. This transformation is in the form of complex Fourier coefficients; thus, we applied an inverse Fourier transform to get the data back in its original form. Now we have noise-free data, and below is a plot that depicts the same. As we can see in the Figure 2.4,

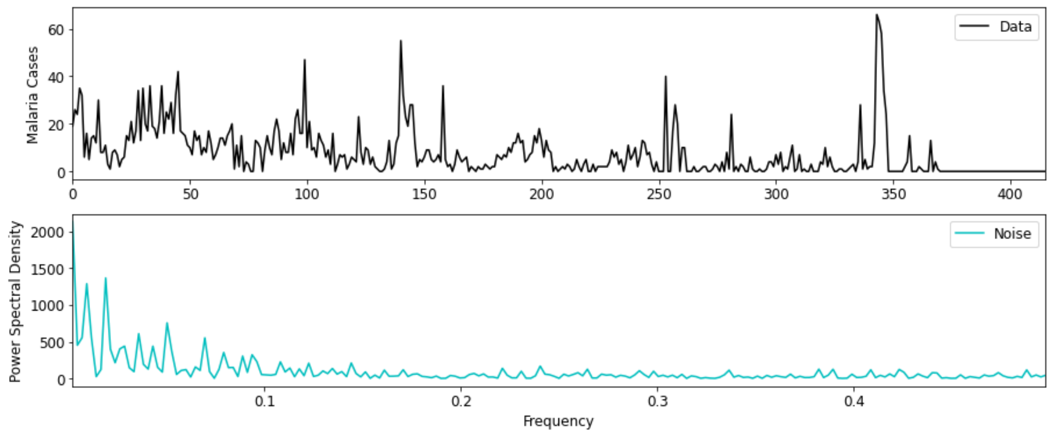


Figure 2.3: Power Spectral Density of Patna Malaria Data

the black curve is our original data, and the orange curve is the noise-free data. At the bottom, we have PSD of both noisy and filtered data, and it is clear that filtered data has exceptionally few dominant frequencies.

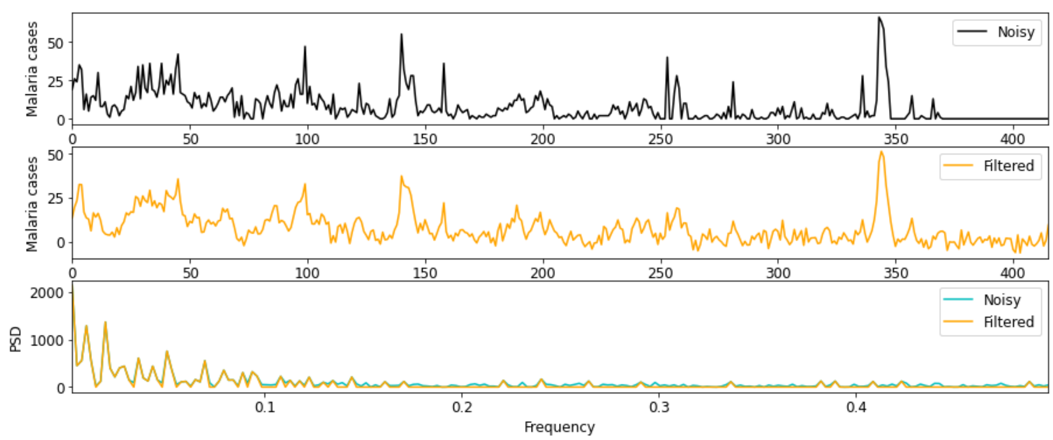


Figure 2.4: Fourier Transformed Malaria Data of Patna

2.6 Model Building

This work seeks to determine how we can model our forecasting algorithm using the above processed data. Many regression and neural network algorithms are tested for that purpose and we try to determine which ones are best predicted in this particular case of use.

2.6.1 Linear Regression

Starting with the simplest model, we tried to regress the meteorological parameters against the health parameters or meteorological parameters against meteorological. Using regression we tried to predict the value of dependent variables based on the values of independent variables using a line that best represents the statistical relation between the predictors and target features. We also introduced a time lag (in weeks) in the meteorological parameters. The hypothesis of behind time-lag is that as we increase the time lag and try to test on the test data then the accuracy parameter will decrease. The equation of linear regression is:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \ldots . w_n x_n$$

Linear regression models try to optimise the above-mentioned equation, that optimisation has to happen based on particular criteria called cost function, which is given by the equation

$$1/(2 * n) \sum_{i=1}^{i=n} (y_i - \hat{y}_i)^2$$

Implementation of linear regression carried out using Tensorflow in python. We used the linear regression for rainfall prediction (based on 40 years of data) and malaria predictions(based on 8 years of data). We haven't used the Fourier transformed data for malaria prediction. A normalised and generator based data used to make predictions in both cases.

2.6.2 Self-Organising maps

Self Organising Maps or Kohenin's maps[32] are unsupervised neural network algorithm. It doesn't rely on conventional backpropagation with SDG; instead, it uses competitive learning to update the neurons' weights. They use a neighbourhood function to preserve the input space's topological properties and produce a low dimensional map of the input space of the training samples, called a map. It is thus used in dimension reduction and uncovering the correlation between data.

Architecture

Self-organising maps have just two layers, an input layer and the output layer or the feature map. SOM doesn't require an activation function in neurons, and thus, we directly pass weights to the output layer. Each neuron in a SOM is assigned a weight vector with the same dimensionality d as the input space.

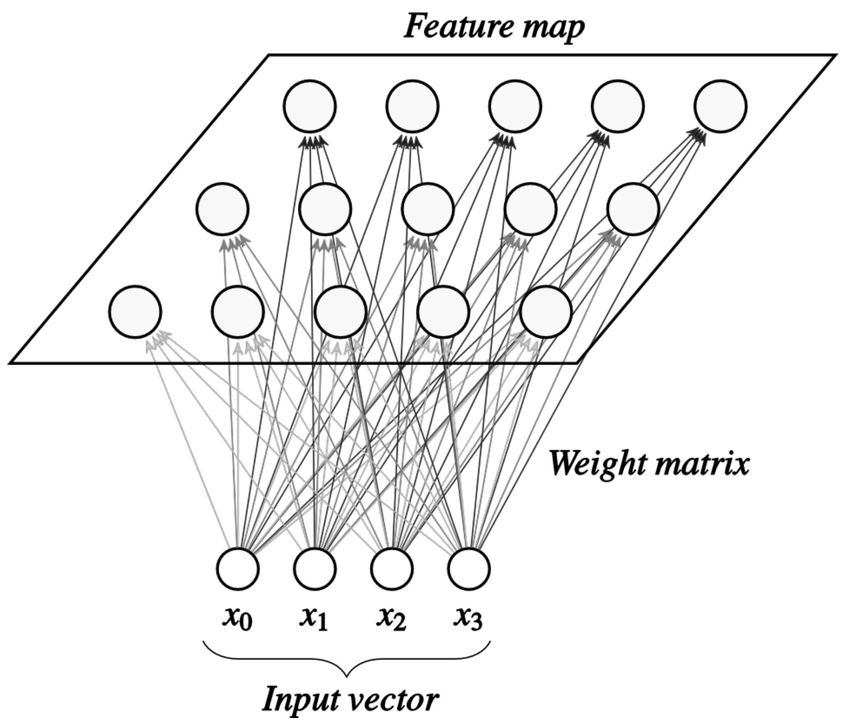


Figure 2.5: Architecture of a SOM network. Source: Github.io[33]

Algorithm

- **Unsupervised SOM**: Unsupervised version of SOM can be used for clustering tasks which after being trained can generate the best matching units(BMUs) for every input. Input of this flow is a vector of length five containing rainfall, humidity, Tmax, Tmin and malaria week-wise. A flowchart of algorithm has been detailed in Fig. 2.4.

- **Supervised SOM**: This supervised SOM can be used for either regression or classification tasks. In our case, a regression task has been applied and the flowchart in Fig. 2.5 conveys the same. The input of the supervised SOM is the same as the output obtained from the Unsupervised SOM. So this is not a pure supervised algorithm as it uses product of unsupervised task as an input.

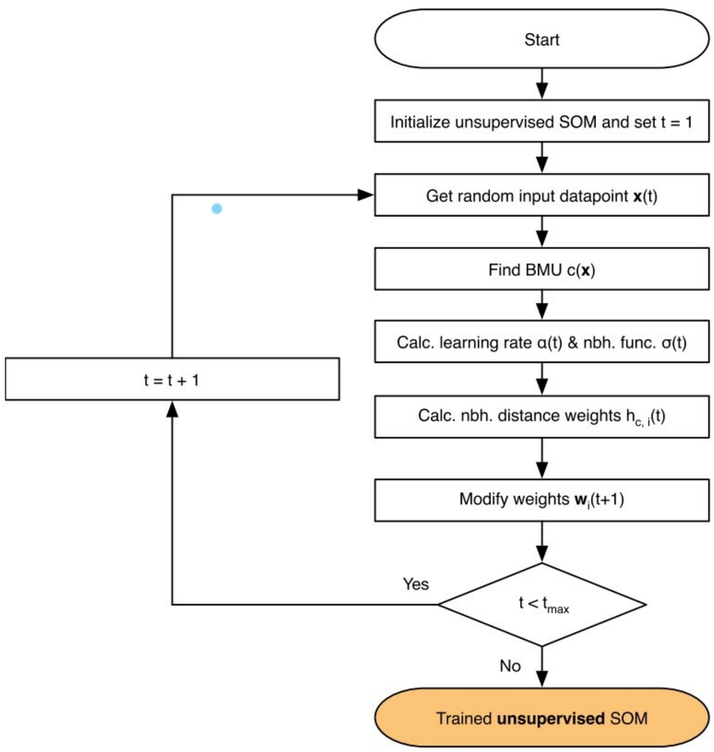


Figure 2.6: Flowchart of the unsupervised SOM algorithm resulting in the trained unsupervised SOM (orange).[34]

Best Matching Unit is a technique which calculates the distance from each weight to the sample vector, by running through all weight vectors. The weight with the shortest distance is the winner. There are numerous ways to determine the distance, however, the most commonly used method is the Euclidean Distance, and that's what is used in our implementation

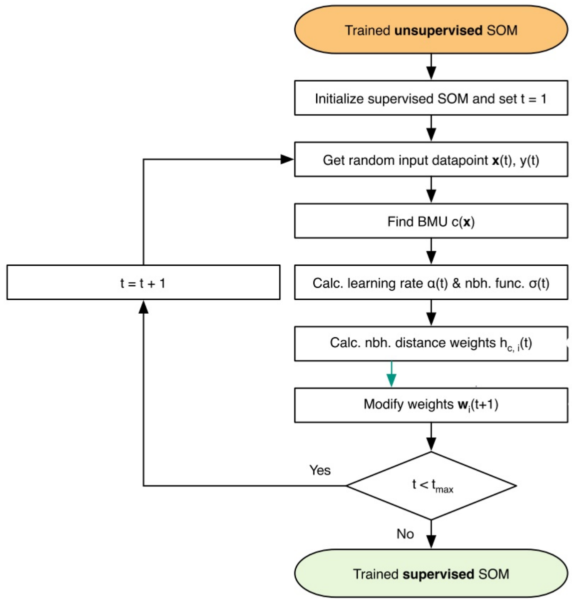


Figure 2.7: Flowchart of the algorithms for the regression SOM. The "Trained unsupervised SOM" (orange) is the result of the unsupervised SOM algorithm illustrated in Figure 2.4[35]

Implementation

There are many python based libraries available to serve the purpose of implementation. I have used SomPy and SuSi package. The predictand for the regression task is malaria with time-lead of 1 week.

- **SomPy**: Sompy is an open-source repository available on github[36]. In this study, this implementation used for dimensional reduction. Each weeks is treated as a dimension and then clustered for each variables. This will show the temporal variation of data over time.

- **SuSi**: This python package for unsupervised, supervised and semi-supervised self-organising maps (SOM)[34][35]. We used this for SOM regression.

2.6.3 LSTM

A recurrent Neural Network(RNN) are a type of Neural Network where the current state has input from the output of the previous step. If we feed a long sequence, they'll have difficulty carrying information from earlier time steps to later ones. During backpropagation, recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update the weights of a neural network. So in recurrent neural networks, layers that get a minor gradient update stops learning. Those are usually the earlier layers. So because these layers don't learn, RNN's can forget what is seen in longer sequences, thus having a short-term memory. LSTM are the solution to short term memory. They have internal mechanisms called gates that can regulate the flow of information. A general LSTM cell has been represented in Fig. 2.8.

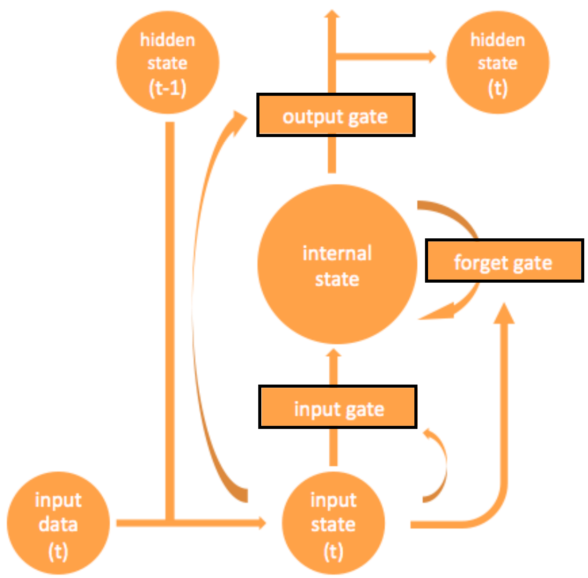


Figure 2.8: Representation of LSTM cell. Source: On the Origin of Deep Learning[37]

- The **input gate** and **forget gate** determine how the next internal state is influenced by the input, and the last internal state respectively.
- The **output gate** determines how the output of the network is influenced by the internal state. Sometimes, it's helpful to remember important details but only use them later.

Architecture

This is a vanilla LSTM model with just 2 sequential layers. First layer is a LSTM layer with 32 nodes and has a input shape of number of variables and window length of 4. ELU (Exponential Linear Unit) used activation function. The second layer is a dense layer with one unit.

- **Activation function - ELU** : Exponential Linear Unit(ELU) is an alternative to ReLU activation function. It tend to converge cost to zero faster and produce more accurate results. ELU approaches smoothness slowly until its output reaches a parameter α ulike ReLU which sharply smoothens.
- **Optimiser - Adam**: It is combination of two gradient descend optimisation techniques: momentum and Root mean square propagation (RSMP). Momentum accelerates the learning algorithm while RSMP is a adaptive learning algorithm. Adam has 3 main hyperparameters: decay rate of the two methods ($\beta_1 \& \beta_2$) and a step size parameter or learning rate α
- **Loss Function - MSE**: The mean squared error(MSE) tells us how close a regression line is to a set of points. It does this by taking the squared distances from the points to the regression line. The squaring removes any negative signs. The smaller the mean squared error, the better the fit or model.
- **Metrics - MAE**: The mean absolute error(MAE) measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables.

Hyperparameter Tuning

As we have discussed, in the LSTM network, few gates and nodes define the flow of data. As the data flows, we calculate the derivatives and cost functions, and at the end of the training, we get the associated coefficients which we roughly call the model parameters. So during model training, only parameters are calculated and updated, but a few parameters are not directly derived from gradients but directly affect the model. These 'extra' parameters usually defined before the learning begins are called hyper-parameters. Finding a perfect hyperparameter for the specific model and data set is very difficult as few of these are very sensitive to initial conditions, and thus a tiny change in their values may lead to a very different model learning.

In our LSTM model, we have adam hyperparameters, learning rate, number of units in LSTM, different activation functions, number of epochs, and batch size as our hyperparameter space which we intended to tweak and find the best optimal value. We used the hyperparameter optimisation technique called grid search. In this algorithm, we define a hyperparameter space for each of the hyperparameters, and then this algorithm uses permutations to try out models on each combination.

	Hyperparameter Space	**Accepted Hyperparameter Value**
Units	16 , 32, 64	32
Learning rate	$10^{-2}, 10^{-3}$	10^{-2}
Activation Function	tanh, ELU	ELU
Epochs	50 , 100	50
Batch Size	32 , 64	32
Beta 1 (Adam)	0.4 - 0.8 (+0.05)	0.50
Beta 2 (Adam)	0.70 - 0.95 (+0.05)	0.90

Table 2.4: Hyperparameter Tuning Space and Hyperparameters values for LSTM Model

The implementation of this grid search is done via Keras grid search. The grid search computes the metrics defined at each combination, and at the end of grid-search, we get the best model. The choice of model is manual. We took the values and metrics for each combination in an excel file and manually found the model with optimal performance on training and the validation set. Keras grid search can automatically output the best model, but this generally considers the better performance on the validation set, which sometimes leads to over or under-fitting.

Model Compilation and Callbacks

After the model compilation using the hyperparameters obtained via grid search with loss and metrics, we have moved to model training which we will discuss in the next section.

Before model training, we have made some callbacks that get triggered only when certain conditions are met.

- **Early Stopping:** This callback can stop the model training at a desired loss or metrics value. In our implementation, if the validation loss value does not change over five epochs, then model training will stop. This method can save us time and extra computing power.
- **Model Checkpoint:** This callback saves the model weight at the regular time interval. The model weights can then be used to forecast the cases where live training is not ideal, like our dashboard(discussed later)

2.6.4 Other(ESN)

We have also tried Reservoir computing based Echo-state network (ESN) but failed due to sensitive dependence of parameters on the initial conditions which is leading towards a chaotic system and hence making it difficult to hypertune the parameters. Reservoir computing is an extension of neural networks in which the input signal is connected to a fixed (non-trainable) and random dynamical system (the reservoir), thus creating a higher dimension representation (embedding). This embedding is then connected to the desired output via trainable units[38].

2.7 Model Training

We have moved to the final phase of our prediction problem, training the model and getting the predictions on the test data. So we have three models, Linear Regression, SOM Regression(used only for rainfall and malaria prediction) and LSTM (Regression and Classification approach). We will discuss each of the models separately.

We will train each model and problem for a four week lead time starting from 1 week lead time. After the training, we will predict and calculate three basic skill scores:

- **Mean Squared Error (MSE)**: The mean squared error tells us how close a regression line is to a set of points. It does this by taking the squared distances from the points to the regression line. The squaring removes any negative signs. The smaller the mean squared error, the better the fit or model.

- **Pearson Correlation**: It is a measure of linear relationship between the two sets of data. It's value lies between -1 to 1. If the value of correlation between two sets is 1 then we can say that the the relationship is perfectly linear and points lie on a straight line for data plots. It is sometimes called Pearson's r.

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2(y_i - \bar{y})^2}}$$

- **R - Squared Value**: We have used coefficient of determination which is r^2 value. Here we have simply calculate it as the square of Pearson's r.

We expect our error to increase, and the correlation value will decrease as we try to look deeper into the future or increase the lead time. Our results section will plot a diagram with error and correlation value on the opposite y-axis and lead time as the x-axis. This plot will be our verification for the model performance.

2.7.1 Forecast Verification

So far, we have assessed our model based on simple statistical measures, which give validation over test data. Now we need to verify our model to determine the predictive power. The verification of the model will give the value of the forecast against some baseline. We will use our test set as the baseline. Model verification estimates how uncertainty in our model prediction will change based on the prediction range.

Since we have two approaches for our model development, series and categorical, we will use two types of model verification methods. We will use the Heidke score for series prediction, and for categorical predictions, we use the RoC curves (details following) for each time lead.

- **Heidke skill score(HSS)**:Heidke skill score(HSS) is generally associated with categorical data, but we have used it in the series prediction. We have created bins of predicted data and then used it as categorical adaptation.

HSS measures marginal improvement of predictions over standard predictions. Measures the percentage of correct predictions, excluding those that would be correct due to pure coincidence. This is a form of generalised skill score where the numerator score is the number of correct predictions, and the reference prediction is a random chance. HSS ranges from $-\infty$ to 1. A negative value indicates that the random prediction is better than the prediction, 0 means no skill, and a perfect prediction gets an HSS of 1.

We have created bins of continuous predictions in a moving window. For example, for rainfall prediction, our 1st bin will be 0 to max predicted values, 2nd bin will be 2.5 to max, then 5 to the max till max. If the predicted value lies in the bin range, that would be allotted 1 and other values outside the bin are 0. This will create categorical data, and then we will calculate the contingency table and then HSS for each bin. This will give the sense that our prediction model is working best for which data range.

If N is the number of samples then N is given by

$$HSS = \frac{(hits + correct\, negatives) - (expected\, correct)_{random}}{N - (expected\, correct)_{random}}$$

$$(expected\, correct)_{random} = \frac{1}{N}\big[(hits{+}misses)(hits{+}false\, alarms){+}(correct\, negatives{+}misses)(correct\, negatives{+}false\, alarms)\big]$$

- **ROC Curves**: ROC (Receiver Operating Characteristics) curve is one of the most efficient and essential metrics to verify the classification problems. The different curves of the ROC curve give the prediction model's proxy of ability to separate between different classes. Higher the AUC, the better the model's ability to distinguish between classes.

  In our case, we will show the ROC curve for week one lead time only(since it is established that performance will eventually decrease for later lead times), and we have three classes as "high risk", "medium risk", and "low risk".

  An excellent ROC has a value close to or equal to 1. It uses sensitivity, specificity, precision and recall to calculate the ROC curves. We have used the sklearn library to plot ROC curves with its built-in modules and plotted using the matplotlib library.

2.7.2 Linear Regression and SOM Regession

We begin our linear regression training process in tensorflow amd SOM regression in SuSi pipeline. The basic difference lies only in the model building process and rest of the processes remains the same. From the implementation point of view in python, we have used object oriented programming to create class of prediction objects.

To summarise, we have used meteorological data like Rainfall, temperature, humidity and soil moisture values to forecast/predict rainfall or malaria values four weeks in advance. This prediction process has been modelled with regression models like Linear and SOM, including splitting, normalising, and lagging the data. The model has been trained and tested, and predictions were generated along with model statistics. The complete workflow has been visualised in the figure 2.9

2.7.3 LSTM Models

We have an LSTM AI model that serves the three different tasks (rainfall prediction, malaria prediction, and temperature window forecasting), but each model's pre-processing and prediction techniques are different, which we will discuss further.

A quick discussion on model performance visualisation

We have six plots (4 graph plots + 2 text segments) for every case. The first plot is the training vs validation loss with the increasing epochs for lead time 1-week forecast. The second plot is the same as the first one but contains lead 2-week data. The third plot is a sort of model validation plot in which the R-squared and MSE are plotted on a common x-axis of lead times, and we are expecting a cross-shaped plot. The following two are just texts fields for depicting metrics values. The last plot is based on whether we have a categorical or series setting (discussed above). If the setting is series, we will see actual vs predicted values on the test data set, and if other settings follow, we have a confusion matrix for three classes. A representation of the above has been shown in fig 2.10.

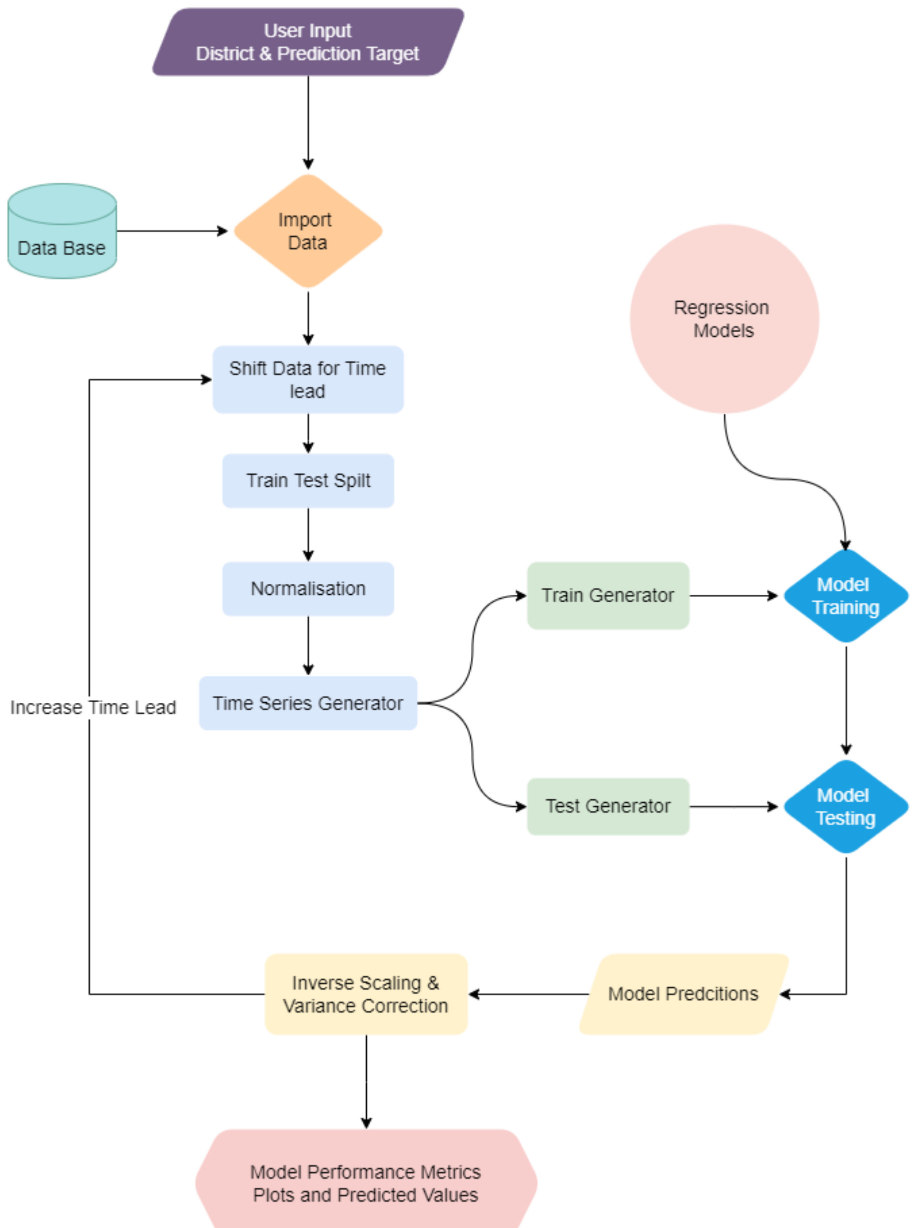


Figure 2.9: Flowchart of the regression problems training

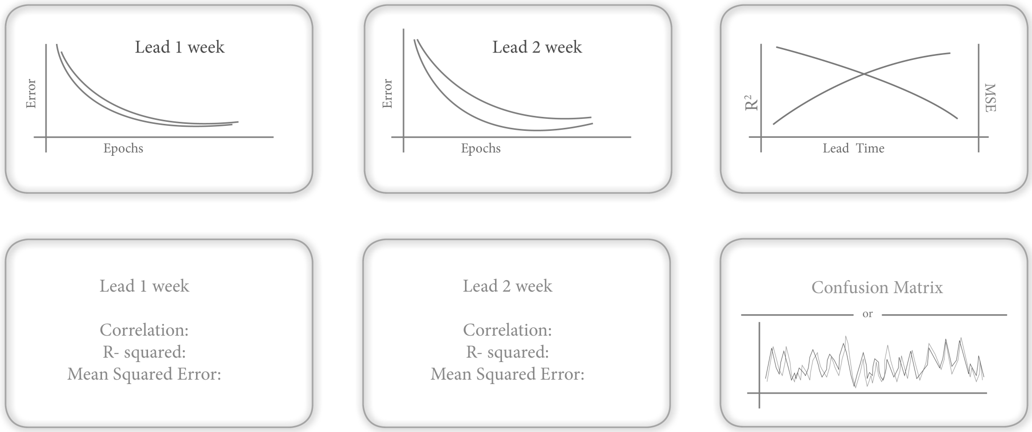


Figure 2.10: A common model performance plots schema for different models

Variance Correction

We have observed a smoothened curve in the actual vs predicted plot for the predictions. To correct this, we have used variance correction to the predicted data. We can implement variance correction as:

$$X_{Var\,Corr} = (X - \bar{X}_{pred}) \cdot \frac{SD_{actual}}{SD_{pred}} + \bar{X}_{actual}$$

Rainfall Prediction Problem

The rainfall prediction problem has been solved using categorical and series approaches. Series uses the raw data while categorical uses the tercile based multi-class rainfall data as predictand. The train-test-split, normalisation and time series process is standard for both approaches. Series and categorical trained on same LSTM model with identical hyperparameters (grid search provides a slightly different value for hyperparameters, but we stick to an identical model for simplicity). The training process has been looped over four weeks lead time by shifting the data. After the 1 and 2 week lead time, we saved the model loss values generated over 50 epochs to plot the loss vs epochs graph. Similarly, we calculated the basic statistics like MSE, Pearson correlation and R-squared value at the end of each loop and saved them in individual lists. We calculate the confusion matrix for the test set for the categorical approach. The predicted values on the test set have been saved in a data frame along with actual values after each training session. The training flowchart is shown in fig 2.11.

This method has also been used to predict malaria with predictand as malaria, and everything kept the same. The results chapter shows that this method has generated predictions with very low correlation values due to noise in the malaria data. We have corrected this noise by using the Fourier transformed data.

Malaria Prediction Problem

We have only used the series data in this malaria prediction problem, which then filtered for noise reduction using the Fourier transform. The period of data is eight years, with six years for training and the rest for testing. Then the process is the same as a series of rainfall predictions where the transformed data is fed into the model over a loop of 4 weeks lead time, and at the end, basic statistics gets calculated along with actual vs variance corrected predictions plot of test data. The workflow of this training is depicted in the figure 2.2

Temperature Window based malaria prediction

In this technique, our first goal is to predict the tmax and tmin values, then categorised based on the predefined window. The process of this pipeline is similar to rainfall except for a few differences in the target set, like we have two variables, tmax and tmin, instead of just one in rainfall prediction. This setting of temperature forecasting is series based forecasting in which we are predicting the continuous values. After the prediction, malaria severity levels get assigned based on the temperature window. Like rainfall prediction, we calculate the error vs epochs curves and metrics based on predicted tmax and tmin values.

After the Tmax and Tmin prediction, we have post-processing of data based on table 1, and a malaria severity has been assigned to each predicted value of the test set from 2011-2020. The malaria cases have been depicted for any particular year in bar graphs up to four week lead times. The plots are shown in the results chapter. Since the malaria levels have been assigned for each district and week, we have generated a malaria spatial map for each week. This visualisation makes more sense in our dashboard, where users can pick any district or week and get the desired results/plots.

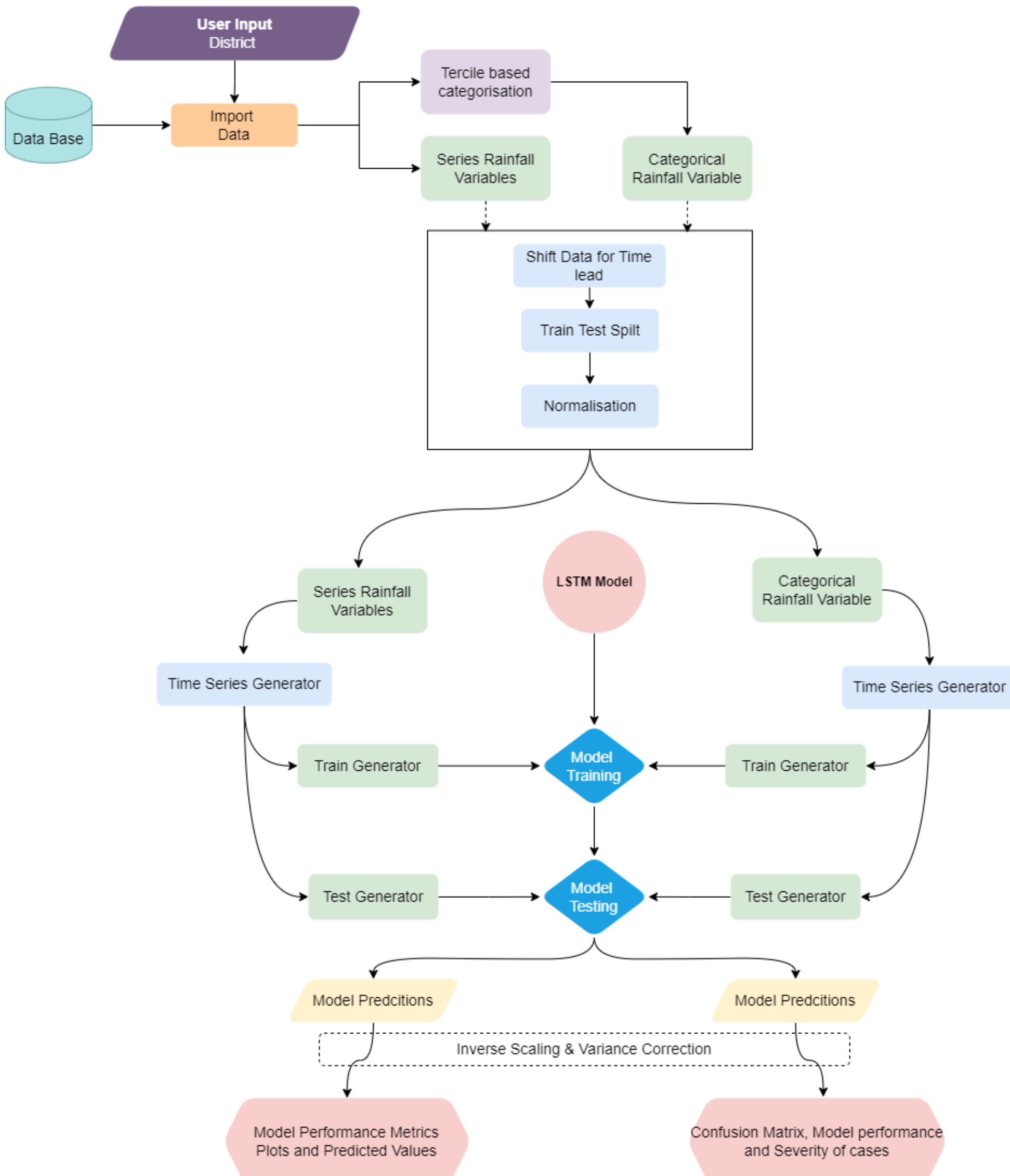


Figure 2.11: Flowchart of the LSTM based Rainfall prediction algorithm

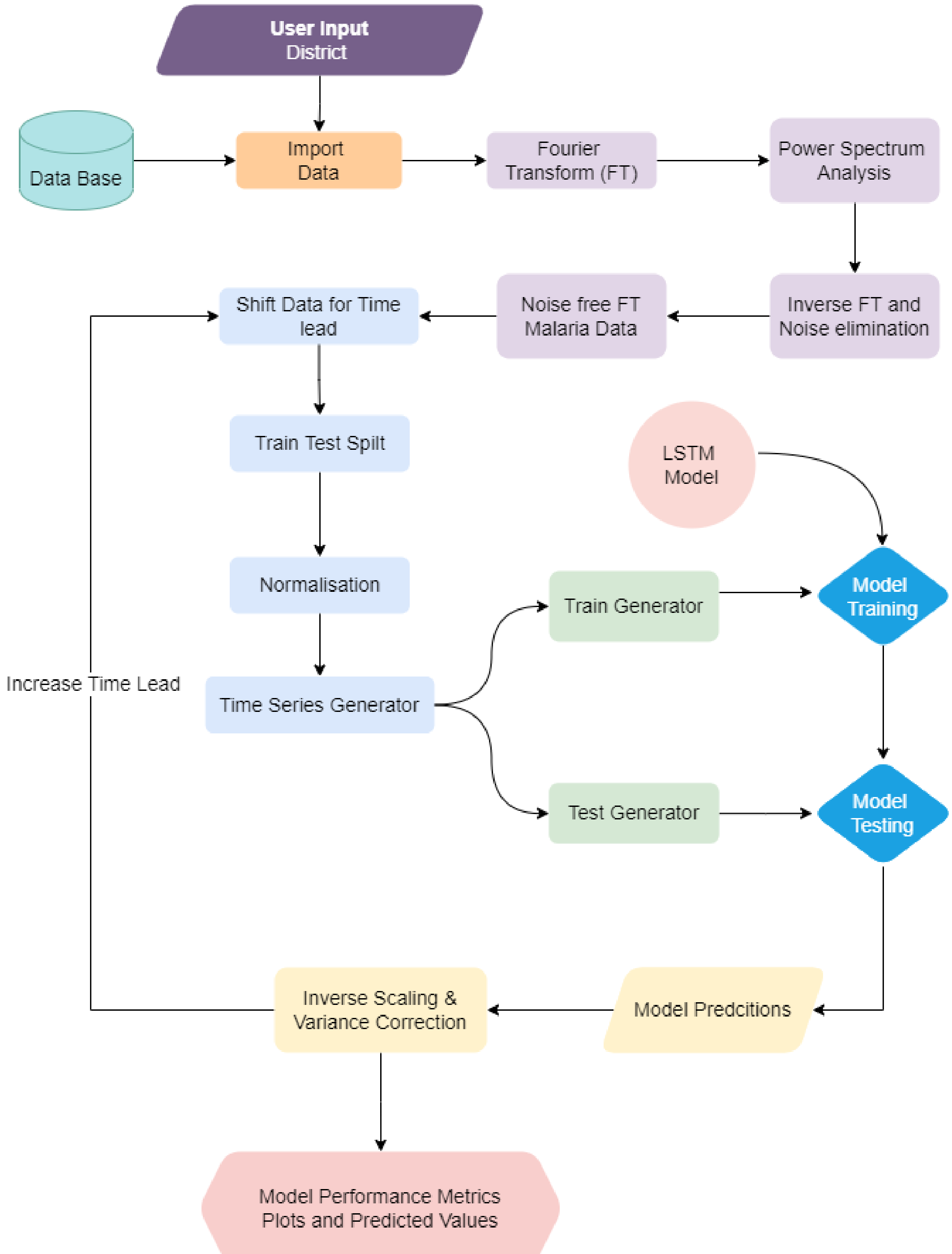


Figure 2.12: Flowchart of the Fourier Transform based malaria prediction algorithm

2.8 Data and Model Dashboard

We have also prepared a dashboard using plotly dash in python. Plotly dash provides an environment for building and hosting a dashboard over web using local server. In this multipage dashboard, we have several features like data exploration for all variables, SOM clustering and it's correlation for each cluster, Linear and SOM regression comparison and LSTM models for rainfall, malaria and temperature window forecasting. There is also a feature of live training and prediction based on the user input for temperature window and malaria forecasting.

Chapter 3

Results

We will discuss how our prediction algorithms for each of the models worked out. We will do this in a linear fashion with starting from rainfall prediction to temperature window based malaria forecasting model.

3.1 Rainfall Prediction

We have three models in the rainfall prediction: Linear regression, SOM regression, and LSTM models. Regression models are based on the series or continuous data, whereas the LSTM model has two variations with series and categorical (tercile based transformation) data. We will discuss each of these individually.

3.1.1 Linear Regression

Linear Regression is the simplest model we have, and we expect our future models to perform better than this model. As discussed in the above section, we have trained the model for up to 4 weeks of lead time and calculated the model statistics. Linear regression analysis for the Patna district is shown in fig 3.1.

In reference to the above figure, we can observe a correlation value of 0.46 for lead time

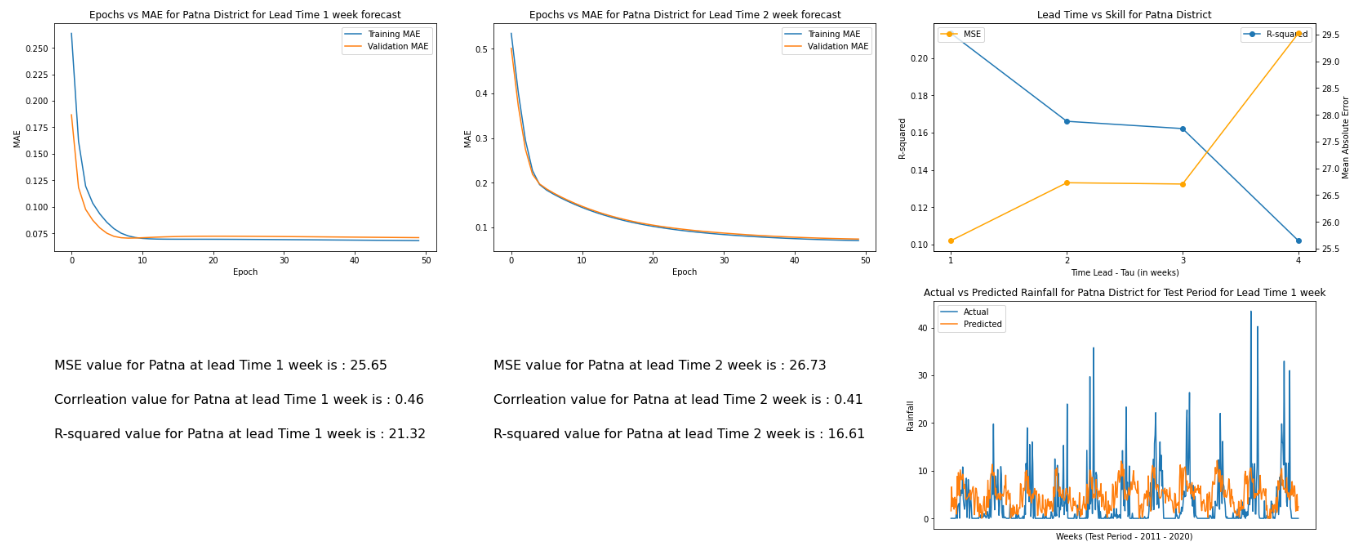


Figure 3.1: Linear Regression performance for Rainfall Prediction for Patna District

1 week and 0.41 for lead time 2 weeks. In plot (1,3), we can see a weak relation of increasing error and decreasing R-squared with lead time increase. A correlation value of less than 0.5 means less than half the variance in the sample is explainable by the correlation. A non-technical sense can indicate a good predictive relationship if one variable is more readily known or gathered. Overall we can say that there is much scope for improvement in model development.

3.1.2 SOM Regression

SOM is not a linear model, so it will introduce a little bit of non-linearity, improving our model performance than the linear regression. We have discussed that our prediction problem is non-linear. Since SOM models are trained using the Susi library, which does not provide modules to save the error on the validation set as the model trains, our plot for SOM will have the epochs vs error plot.

The above plots show a decent prediction model, but there is scope for improvement. The increase in error and decrease in R-squared value as we try deeper into the future holds our hypothesis true. The correlation value is more than 0.5. The predictions are not equalising the spikes in the data. The SOM regression models the overall trend of the data rather

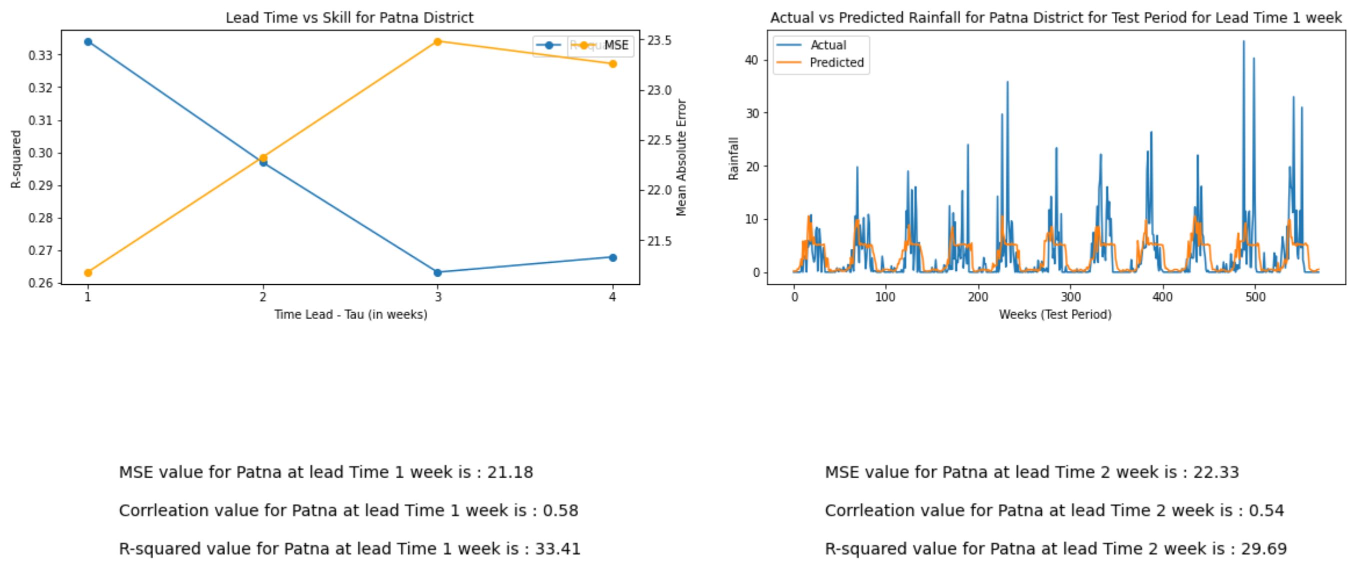


Figure 3.2: SOM Regression performance for Rainfall Prediction for Patna District

than the Crest and trough of the data. This model can not be considered a good model for prediction due to its inability to account for the spikes.

3.1.3 LSTM Model

LSTM model is our state of the art model, which has good predictions scores. LSTM model has been trained for regression and classification tasks which we will discuss individually.

Regression/Series Prediction

This is a series to series prediction problem. We have output as continuous rainfall values. Fig 3.3 shows the Rainfall prediction of Patna district tested over the range of 2011 - 2020.

As we can see, the correlation values are close to SOM Regression, but this LSTM captures the crest and trough of the data in a much cleaner fashion. As expected, the Corr vs lead-time vs error has a very nice cross shape. The learning curve has a slight deviation towards the end of training. Since the correlation value is just close to 0.6, we will try to improve upon that.

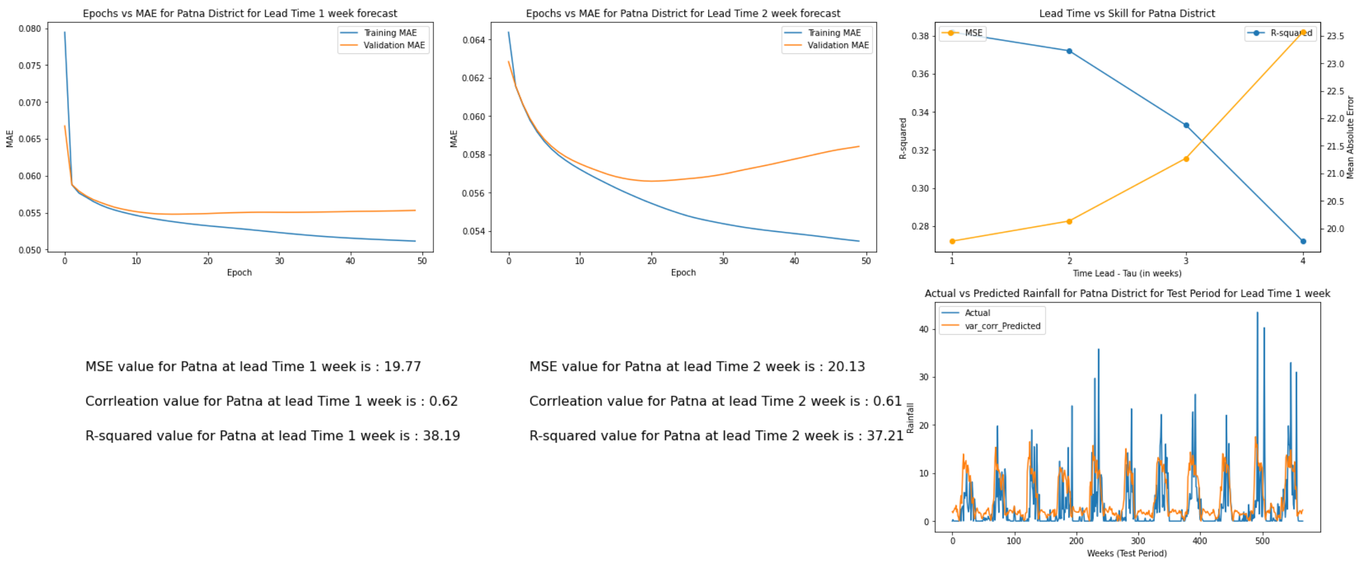


Figure 3.3: LSTM series model performance for Rainfall Prediction for Patna District

Categorical Prediction

We have adopted the categorical prediction method to improve upon the correlation value. The predictand here is multi-class rainfall data. The output of this model is segregated into classes using upper and lower tercile values. We will have a confusion matrix instead of an actual vs predicted plot.

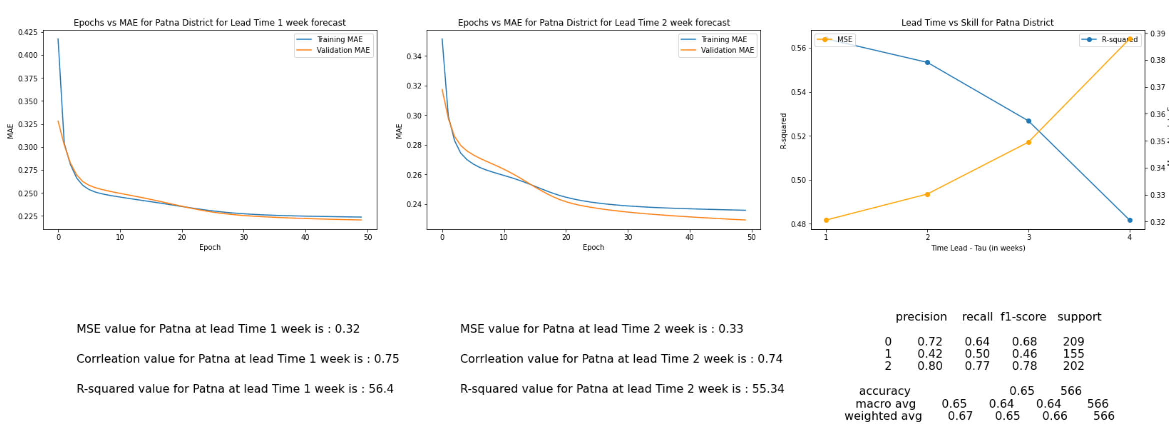


Figure 3.4: LSTM categorical model performance for Rainfall Prediction for Patna District

Looking at the learning curves, it is clear that the categorical adaptation of LSTM has a clear advantage. The correlation value is much higher than the previous case, along with a clear dependence on MSE with lead time. The precision, recall and accuracy of the prediction are also very decent.

3.2 Malaria Prediction

Here we will try all the rainfall prediction models in the previous section. There would be any additional adaptation with the Fourier transformed data in the LSTM model. LSTM model is tested on categorical and series data, whereas linear, and SOM regression with the Fourier transformed data is tested only on series data. We will directly jump into each model performance individually

3.2.1 Linear Regression

This model has been trained for 6 years and tested on 2 years of data, unlike rainfall predictions. The skill scores shown here are only for the test period for Patna district.

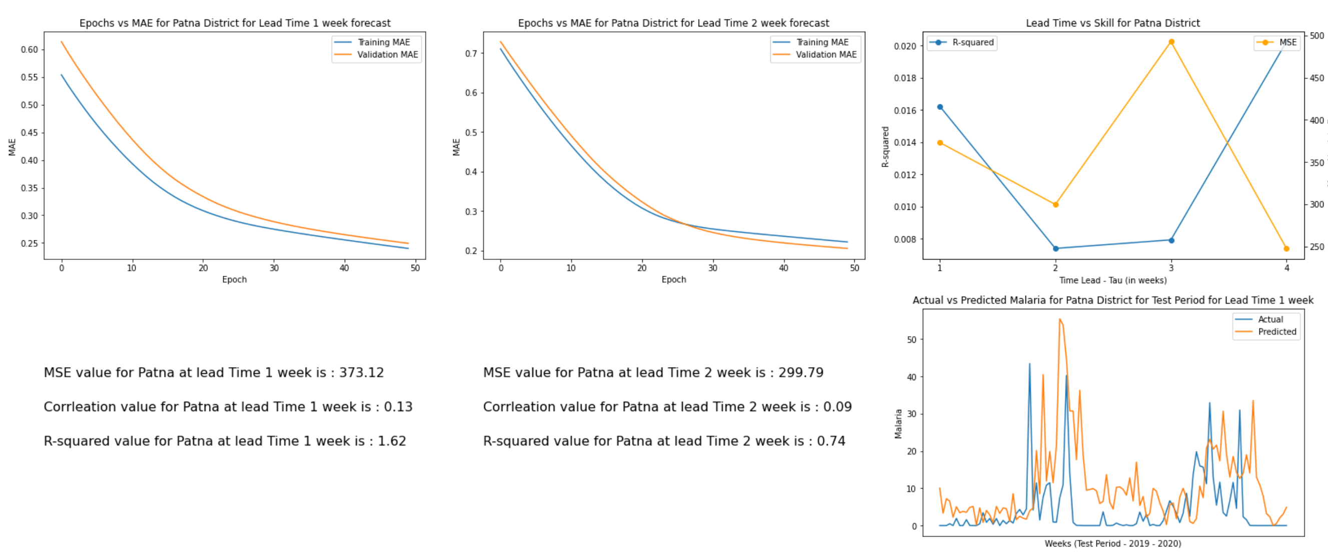
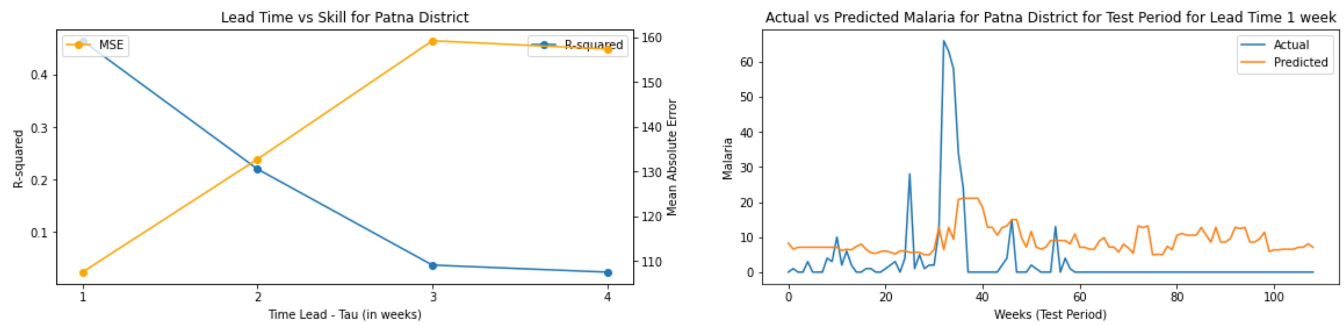


Figure 3.5: Linear Regression model performance for Malaria Prediction for Patna District

Linear regression has an abysmal performance for malaria prediction. The correlation value is close to zero, indicating no linear relationship between malaria and meteorological variables, which holds for our initial hypothesis. Thus a non-linearity is a must for this prediction problem. The training curves for the linear regression are smooth. The R-squared vs time-lead vs MSE plot is chaotic and does not provide meaningful insight.

3.2.2 SOM Regression

As we have discussed above, the SOM has a factor of non-linearity. Thus it should capture the problem better than simple linear regression.



MSE value for Patna at lead Time 1 week is : 107.52

Corrleation value for Patna at lead Time 1 week is : 0.68

R-squared value for Patna at lead Time 1 week is : 46.5

MSE value for Patna at lead Time 2 week is : 132.77

Corrleation value for Patna at lead Time 2 week is : 0.47

R-squared value for Patna at lead Time 2 week is : 22.01

Figure 3.6: SOM Regression model performance for Malaria Prediction for Patna District

Looking at the plots of SOM regression, it is clear that it has a clear advantage over linear regression, but there is still room for improvement. The correlation values are less than 0.5, which suggests that less than half of the variability is explained by this model. We do not have learning curves here due to package constraints. The actual vs predicted plot has many mispredictions over the entire range, ruining the purpose of prediction. In this case, even if the correlation value is close to 0.5, the predictions are very poor. This prediction plot is only for the Patna district, but using our Plotly dashboard (link in the Dashboard section), we can produce predictions for any district, and these results are consistent.

3.2.3 LSTM Prediction

Like the rainfall prediction, we will also try our model in series and categorical mode. The categorical mode will be further explored using the Fourier-based transformations in the following subsection.

Categorical Prediction

Here we have kept everything the same as the rainfall prediction except the data part. We have used only 8 years of data with 6 years of training and 2 years of testing, for which the results are shown.

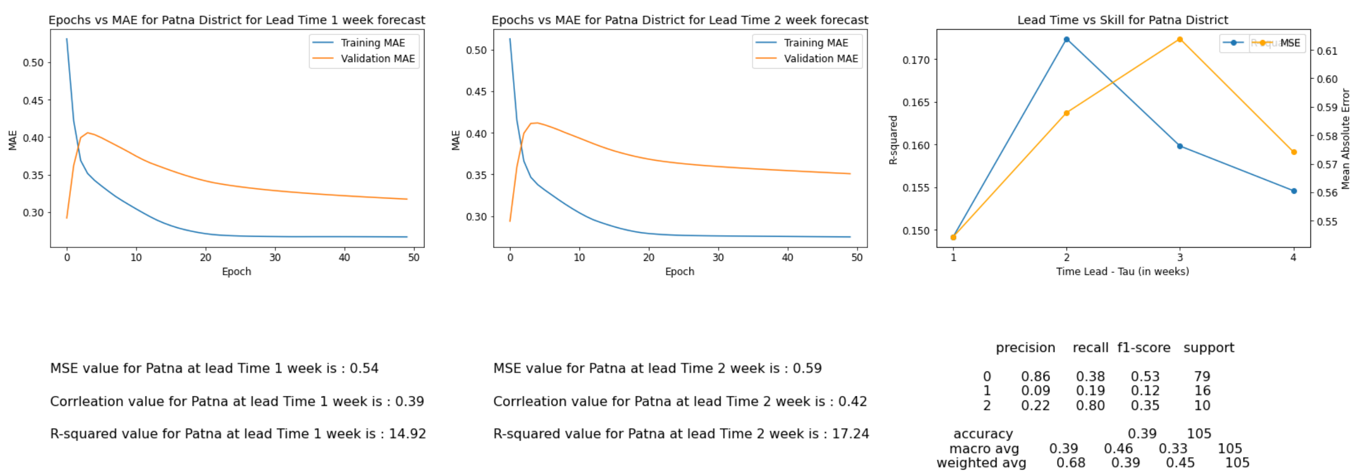


Figure 3.7: LSTM Categorical model performance for Malaria Prediction for Patna District

The results here are worse than SOM regression, and learning curves are clearly showing that the model is overfitting. Hence, the categorical adaptation is overfitted the data and has poor skill scores.

Series Prediction

The same rainfall prediction model has been tested for malaria prediction while keeping the hyperparameters constant. The results here are shown for the Patna district over the test data.

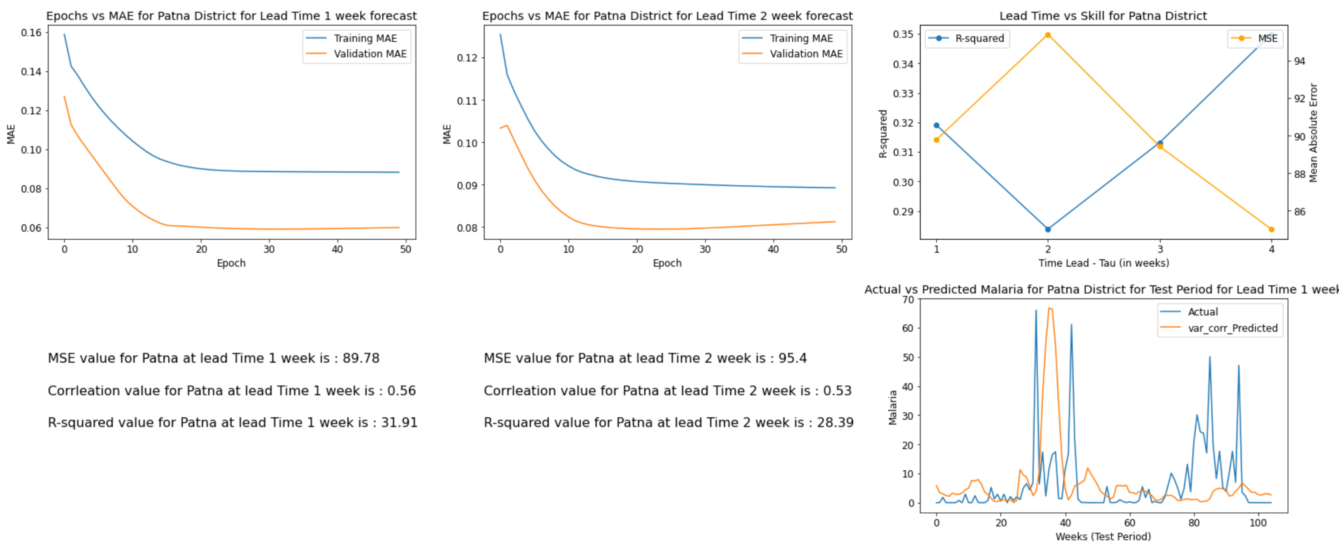


Figure 3.8: LSTM Categorical model performance for Malaria Prediction for Patna District

This is the best model so far we have. It has a clear advantage over SOM regression and categorical adaptation in the terms of correlation value. The problem with this model is that the R-squared vs time-lead vs MSE plot is very unusual as we can see a sudden rise in the value of R-squared and a dip in MSE. The other problem is that the validation loss is less than the training loss. We will discuss the possible reasons in the discussion chapter. We suspect that problem could be due to lack of enough data for a deeplearning model to work properly.

3.2.4 Fourier Based LSTM Model

The fundamental problem in the last few approaches is that we have much noise in the data. Also, the data has a very improper balance between train and test sets due to a lack of data points. We will use Fourier based data and feed into the same series LSTM model and run the same checkpoints, and results are shown here for the test set.

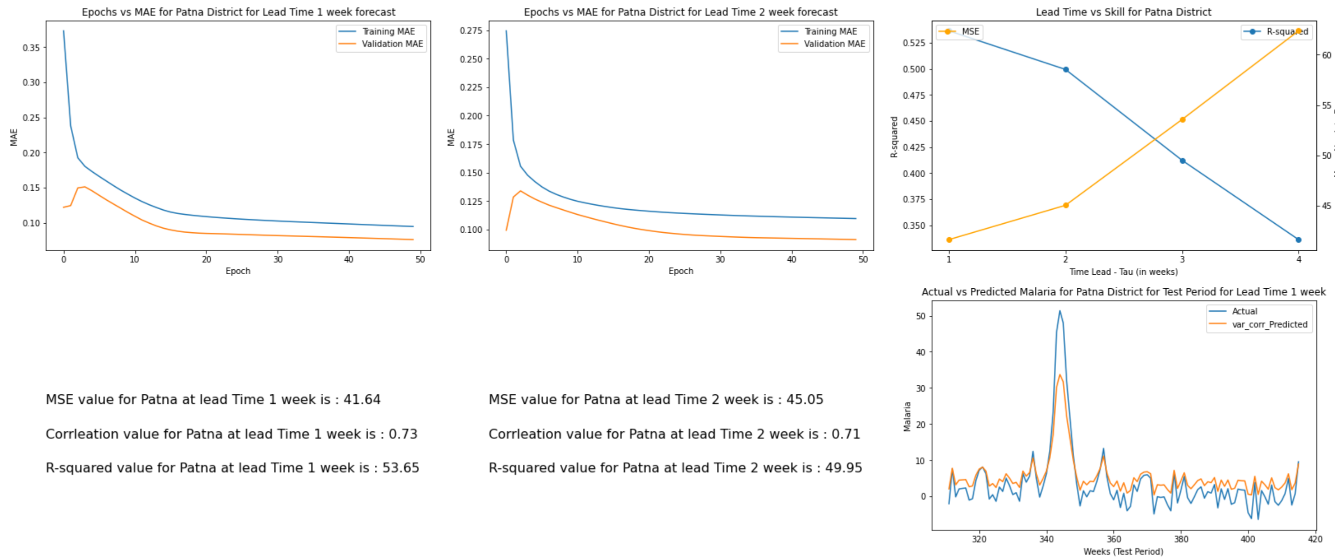


Figure 3.9: LSTM Fourier transformed model performance for Malaria Prediction for Patna District

This model is the best malaria prediction model we have developed. It has a very high correlation value and relatively low error. However, it has a few problems like lower validation loss than training loss. We will address these issues in the discussion chapter. This model also has a nice decreasing R-squared value and increasing error with the increase in lead time.

3.3 Temperature Window based Malaria prediction

This is another malaria cases forecasting algorithm that is solely based on temperature windows. Here we will only account for the Tmax and Tmin values. We will use our LSTM model to predict the Tmax and Tmin values simultaneously, and then based on Table 1, we will assign the malaria severity values.

In the Tmax and Tmin data, there is a clear cyclicity of 1 year. The prediction task would be much easier as the task involves humidity and rainfall, which has almost the same cyclicity.

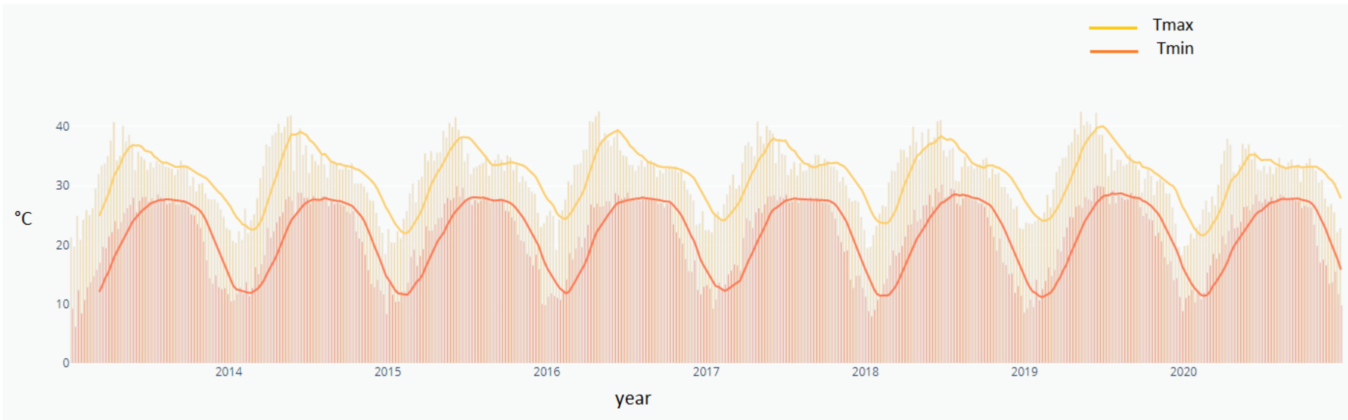


Figure 3.10: Data Trends of Tmax and Tmin for 2013 - 2020 for Patna district

Now using the data for Tmax and Tmin, we have trained the LSTM model for forecasting. Since the output has 2 nodes, there have been slight tweaks in the last layer of the model. Hyperparameter tuning via grid search yields a slightly different value, but the end prediction has no apparent changes, so we have kept the old model. The output of the model for the Patna district is shown below. The mode of training the model is a series.

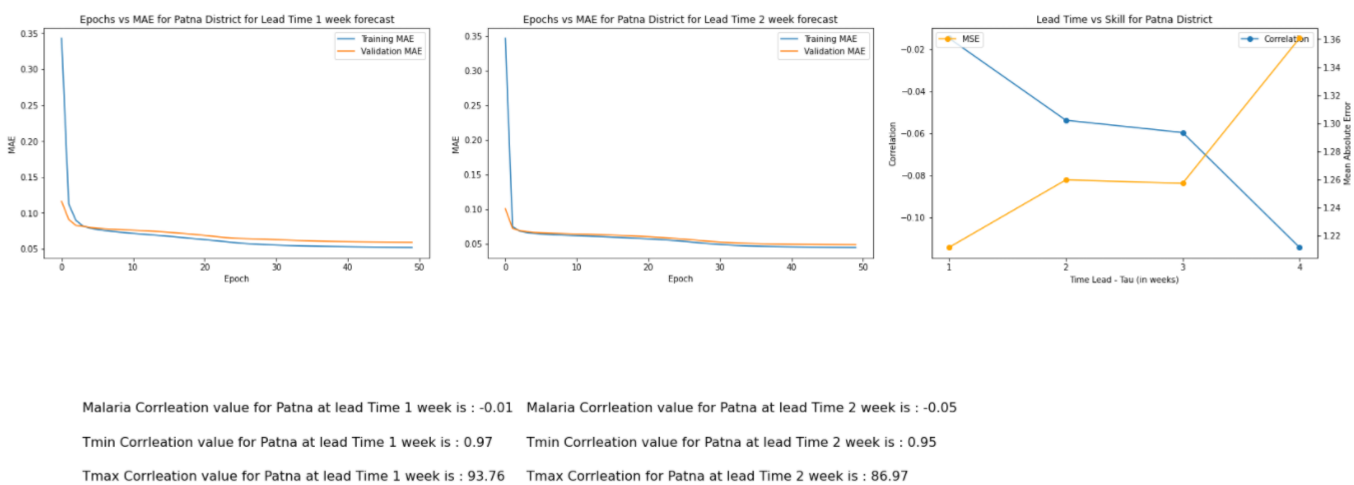


Figure 3.11: LSTM Model performance for Tmax and Tmin prediction for Patna district.

As we can see in the above plots, the model has a very high correlation value of 1. This is due to the continuous cyclicity trend in the data. Also, the R-squared vs error vs time-lead plot has a nice cross shape. The actual vs predicted plots could not be generated as there are two outputs.

These predicted Tmax and Tmin values have been gone through the model post-processing, and a malaria severity level has been assigned based on table 1. A bar graph has been generated for up to 4 weeks lead time. The bar graph for the Patna district is shown below.

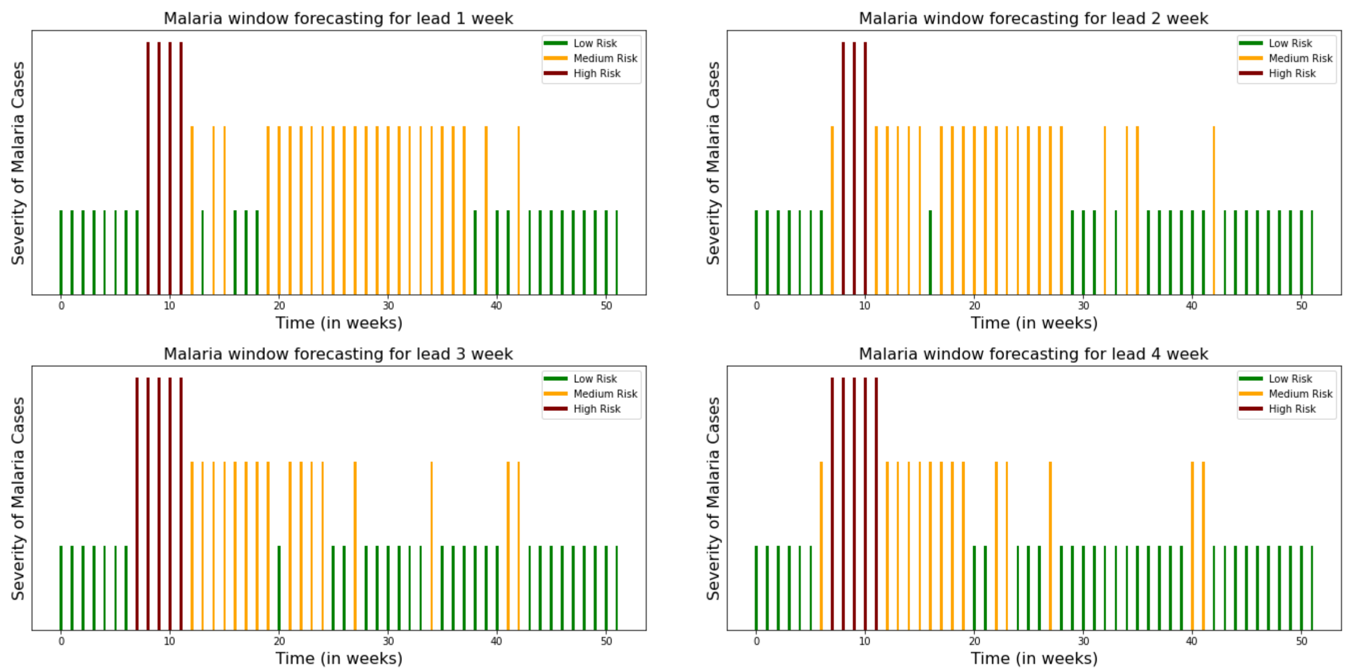


Figure 3.12: Malaria Severity level for Patna district over year 2013 for 4 week lead time.

Now we have the malaria levels for each district of Bihar at each week. So now we can plot the spatial map of malaria cases for any week. Below is the spatial map for 24th feb 2013.

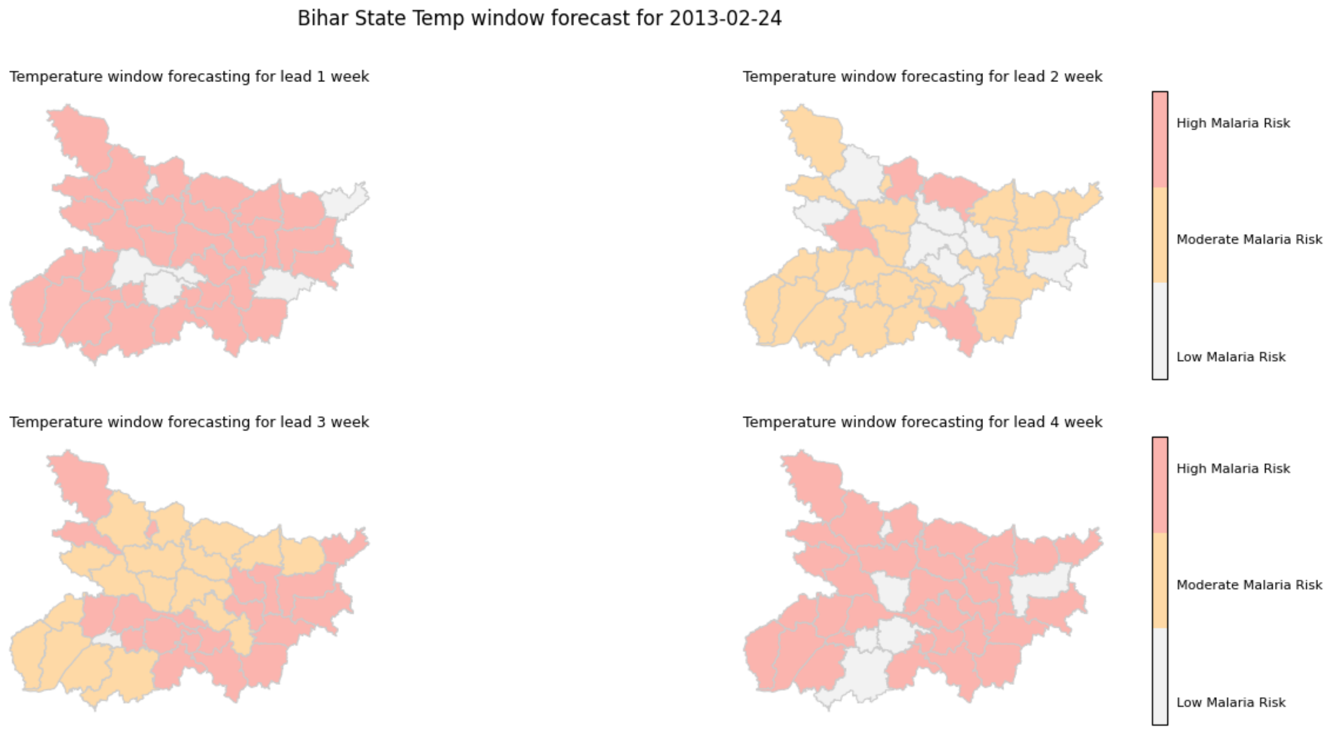


Figure 3.13: Malaria Severity level for Patna district over year 2013 for 4 week lead time.

Chapter 4

Discussion

We will discuss each of our three models individually, starting from Rainfall prediction to temperature window-based malaria prediction.

4.1 Rainfall Prediction

We have tried models with increasing complexity starting from Linear regression to neural networks. The linear regression and SOM regression have not shown very good predictions, although the models do not show any underfitting or overfitting.

Our LSTM model has given very satisfactory results. We have tried two variations, series and categorical, and we get excellent prediction scores for categorical data. Performance of categorical over series arises may be due to the decrease in the variance of predictand. At the core, we perform the regression task, but the predictand has only three values out of the set of integers.

This project aims to forecast malaria cases, but here we are discuss rainfall prediction because if we perform a power spectral density (PSD) analysis over malaria and meteorological data, we observe a common dominant spike in both data trends. This common power will give us a proxy that if we build a model for rainfall prediction, then given enough malaria data, we can also achieve the same level of accuracy. This statement can be thought of as using the PSD, we have tried to uncover the common hidden patterns across data, which can be used to build a general model. Fig 4.1 shows the PSD of malaria and other meteorological data with a common dominant spike.

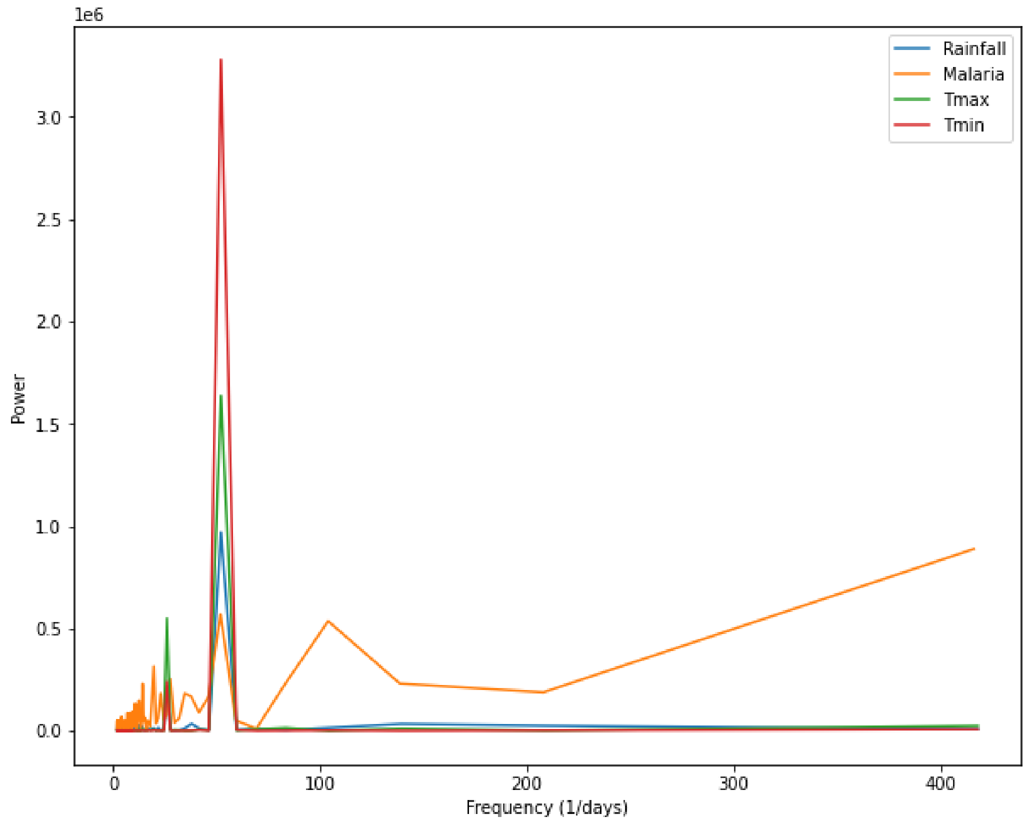


Figure 4.1: PSD of Malaria with meteorological data for Patna District. A common dominant spike is visible.

Since we have enough rainfall and other meteorological data, we have tried to build a model for rainfall prediction and use the same for malaria prediction using fewer data. So far, we have managed to achieve our goal in predicting malaria cases using the models used for rainfall prediction, which we will discuss next.

Figure 4.2 visualises the Heidke skill score for our LSTM Rainfall prediction model. As shown in the plot, Heidke's score is highest for the lead 1-week forecast and lower for the subsequent lead times. Also, the peak for each lead time, we have a peak at Bin 5. So we can say that this LSTM based forecasting model can work the best for the predictions of

5mm of rainfall and above.

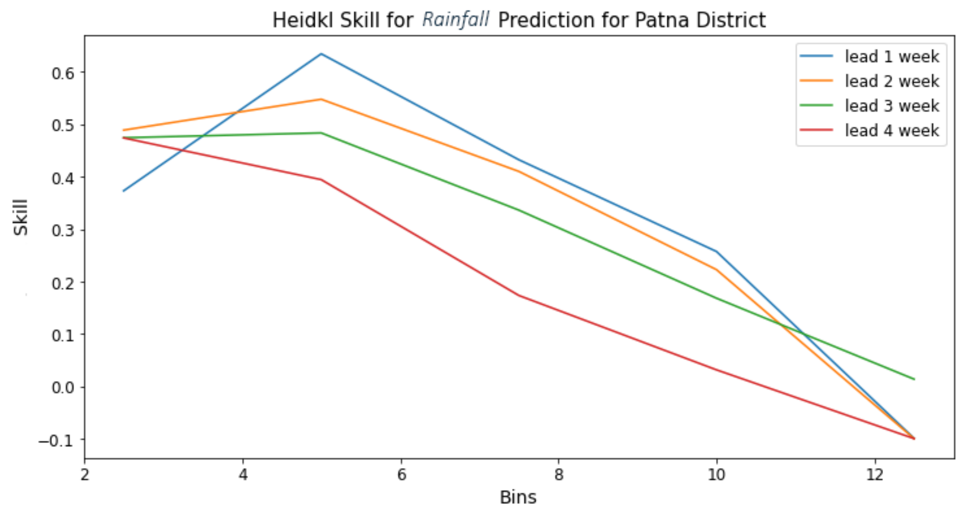


Figure 4.2: Series Rainfall prediction verification using Heidke Skill score for Patna District

We have also tried the Heidke skill score for the categorical method, but the number of classes is restricted to just three, producing very preliminary results. Hence we have plotted the AUC curve instead for week 1 lead time. The ROC curve has been depicted for three severity levels. High risk has the highest area while the medium risk is close to the average, while low risk has the least area. Hence, this categorical model has decent performance in the high to medium range. The same pattern we have observed in the series forecasting where the model is struggling in the lower values but has a decent performance in the higher values. Thus these two types of model approaches complement each other.

Table 4.1 summarises the correlation, MSE and R-squared values for the rainfall prediction over 4 week time leads of all models.

4.2 Malaria Prediction

Malaria transmission was found to be seasonal for a few districts having a good distribution of data. Relative humidity and rainfall have a significant correlation with malaria, as seen in the SOM clusters. The study by Loevinsohn ME has shown similar results[39]. This

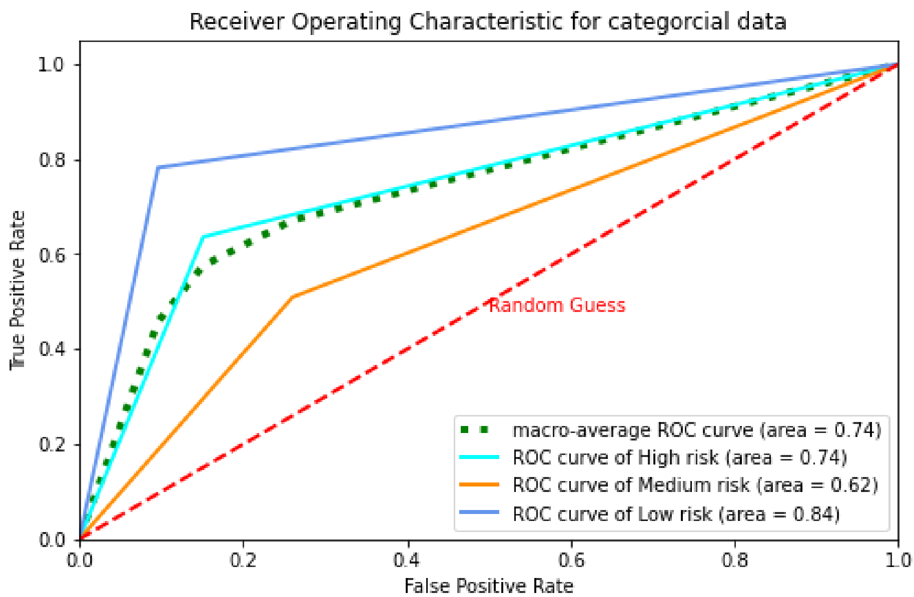


Figure 4.3: ROC Curve for categorical Rainfall prediction with 3 classes for Patna District for week 1 lead time

may be the case due to the increased number of breeding grounds during the rainy season. There is also a decreased number no of cases before and after the monsoon. Temperature also a significant role in malaria transmission. Due to increased temperature, the incubation period of malaria decreases, and hence there is a risk of higher cases. Here, the model's min and max temperature variables have a significant role in malaria development. Apart from the meteorological and health factors, socio-economic factors, urbanisation, health infrastructure, immunity to malaria, and migration have a role in malaria transmission.

Here we have tried the few predictive models discussed in the above section. We have used the same models. The regression model performs the worst. It has very high MSE values for training and validation, which shows underfitting. In chapter 1, we have discussed that this prediction problem is non-linear, and linear regression supports our hypothesis.

Next, we have SOM regression which performs far better than linear regression due to the addition of the element of non-linearity. We also encounter high MSE values and correlation values less than 0.5. So we need to improve our model further by using the LSTM model, which performs the best for the rainfall prediction.

	Correlation				MSE				R-Squared			
Lead Time (weeks)	1	2	3	4	1	2	3	4	1	2	3	4
Linear Regression	0.42	0.40	0.44	0.36	26.23	27.00	25.94	27.89	0.17	0.16	0.19	0.13
SOM Regression	0.57	0.54	0.51	0.51	21.17	22.32	23.48	23.25	0.33	0.29	0.26	0.26
Series LSTM	0.61	0.61	0.58	0.50	19.88	20.04	21.07	24.25	0.38	0.37	0.33	0.25
Categorical LSTM	0.75	0.74	0.72	0.70	0.31	0.33	0.34	0.37	0.56	0.55	0.52	0.49

Table 4.1: Summary of R-squared, MSE and Correlation values for Rainfall prediction

The LSTM for malaria prediction has significantly fewer data. In comparison, our whole malaria data is less than the test set of rainfall prediction. So working with the 20 percent will bring the overfitting or underfitting or even some exciting scenarios.

Since the categorical adaptation of LSTM works best for rainfall prediction, we have tried that first, but looking at the result, we can say that this is a very bad model. We have used MSE as the error function, but using the MSE for categorical prediction is not a good idea as we have fixed variance across the data due to pre-defined class boundaries, so the difference among those classes will not make sense while calculating the MSE. The underperformance of the categorical model can be accounted for the fact that the complexity of the data varies too much between train and test data due to scarcity of data which may have led to the creation of biased partitions for categorical data.

Next, we have the series adaptation of LSTM models, which performs better than the previous one. Since we have two years of the test set, categorisation based on the tercile values might cause unknown biases. The series adaptation does nor face such issues and thus has a better performance.

We have discussed in the results section that, LSTM model has lower validation loss than training. This is unusual and it may happen because the validation set is "easier" than the training set, but this can't be changed because we are using the time series data and taking out data from the middle of series might cause ruining the overall time series data. The other reason might be the lack of regularizers used in the model. The other reason might be the lack of data available for the test/validation set. If we look closely at the test data, there is a peak and the rest of the data is way below the half of peak, this might cause the model to predict only for few features in the test set.

	Correlation				MSE				R-Squared			
Lead Time (weeks)	1	2	3	4	1	2	3	4	1	2	3	4
Linear Regression	-0.17	-0.14	-0.13	-005	285.7	369.4	380.9	238.7	0.03	0.02	0.018	0.001
SOM Regression	0.68	0.46	0.19	0.15	107.5	132.7	159.24	157.38	0.46	0.22	0.036	0.023
Series LSTM	0.54	0.53	0.55	0.59	92.9	94.8	90.95	84.9	0.29	0.28	0.30	0.34
Categorical LSTM	0.404	0.408	0.403	0.36	0.52	0.57	0.59	0.60	0.163	0.166	0.162	0.136
Fourier LSTM	0.73	0.70	0.64	0.57	41.64	45.04	53.56	62.36	0.53	0.49	0.41	0.33

Table 4.2: Summary of R-squared, MSE and Correlation values for Malaria prediction

We have also performed the Heidke skill score for malaria prediction, similar to rainfall predictions. The results are not straight forward as rainfall, where we have observed a constant decline in the skill score with the increase in lead time. Here in the Figure 4.3 , we can infer that the prediction of 10 or more has very good confidence. The overall trend of the time leads is very unpredictable. The reason can be stated due to the test size, which is just 104 points. Also, as we are creating bins, we are dealing with less and less data which will eventually lead to poor prediction expectations.

Since Fourier transformed model was our best model for malaria prediction which is based on series data so we are not comparing the categorical counterpart in this case.

4.2.1 Architecture sensitivity analysis for the LSTM model

So far in this project, we have used a simple single-layered LSTM model. But we need to explore more to check the model performance as we tweak the model's architecture while keeping all the parameters the same.

In our single layered LSTM model, we have 32 neurons(per the hyperparameter tuning grid space). We have tried adding another layer with a decreasing number of neurons by 2.

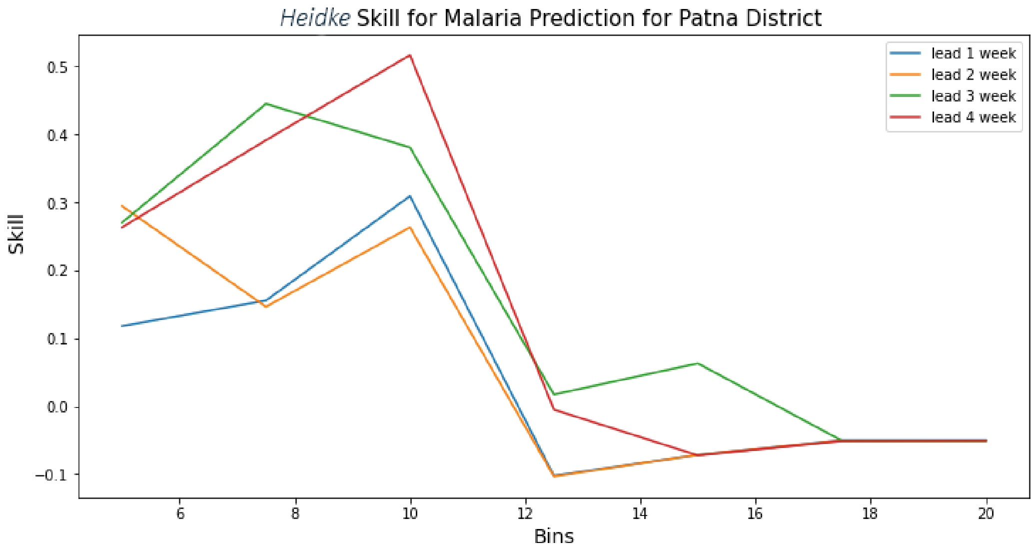


Figure 4.4: Prediction verification using Heidke Skill scores for Series malaria prediction using Fourier transformed data

So we will have an architecture with 3 LSTM layers with 32, 16 and 8 neurons, respectively. We have tested the performance for a few districts, and plots are given below for each district with a lead time of up to 4 weeks.

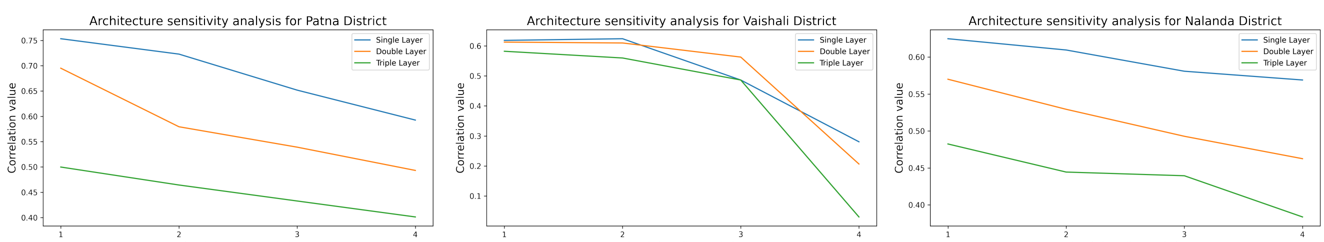


Figure 4.5: Architecture based analysis for LSTM model with increasing complexity

In the plots over three districts, we can see that as we increase the complexity of the model, the overall correlation value on the test data decreases. For most cases, this trend is linear except for a few districts like Vaishali. This may be happening due to the increase in the model's overall complexity, which causes overfitting and thus a poorer performance on the test data.

We have also tried to increase the number of layers while keeping the number of neurons same. We have kept number of neurons as 32 (obtained via grid search). We get the same general trend as the fig 4.5. So we can conclude that increasing the complexity may be not be the good idea as it induces overfitting.

When we talk about memory-based models like RNNs or LSTMs, then have learning hierarchical features layer by layer where starting layers get the bigger picture, and as we go towards the end, layers learn the finer things. We have tried to increase complexity by increasing the dense layers instead of complex LSTM layers, and the results are exciting.

We have added the dense layer in the hierarchy. Earlier, we have a model with an LSTM layer of 32 neurons and a dense layer of just one neuron. We have tried to add more layers in succession by 2. So in the end, we have a model with an LSTM layer with 32 neurons and five dense layers with 16,8,4,2 and 1 neurons, respectively.

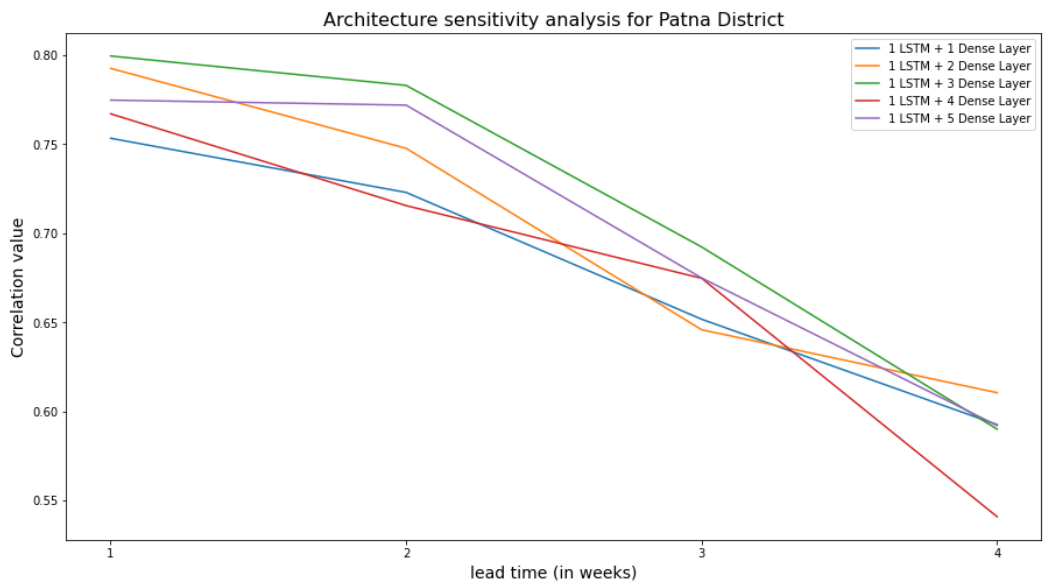


Figure 4.6: Architecture based analysis for LSTM model with increasing complexity in Dense layer

Here we can see an interesting pattern that as we increase the number of dense layers, the correlation value of the model increases but up to a certain threshold. After the 4 dense layers, the complexity increases, and the overfitting comes into play. The LSTM layer doesn't give out a softmax output so we need to provide a dense layer for the c The output of a LSTM is not a softmax but the internal state h as output, so the dimensionality of this output is equals to the number of unit, which is probably not the dimensionality of our desired target. That's why we have to specify a last dense layer, which merely correspond to this equaltion: $y_t = W * h_t$ where y will be the logits we have to pass in a softmax layer, and W the weight connection matrix of this last layer.

So the dense output layers are the ones that are making the predictions, and hence making them too complex would do overfitting, as we saw above. So we can incorporate adding a few extra dense layers to the model, but since we have tried to keep the model extremely simple here, we are just going with a single dense layer with one neuron.

4.3 Temperature Window based Malaria Prediction

Temperature window-based malaria prediction is a straightforward model offered by WHO for malaria prevention across the globe. We have used the same LSTM model and predicted the Tmax and Tmin values simultaneously. Then the predicted values were passed through a filter defined in table 1.1 and assigned the malaria severity levels.

Since this is a very simple yet effective model, we have created a different dashboard where users can input the meteorological values of the past 4 weeks and get the predictions for the next four weeks based on the temperature window.

Figure 4.5 depicts the ROC curve for the temperature based malaria prediction for Patna district for week 1 lead time. Like the rainfall predictions, it has also the bigger area for high risk and lowest area for the low risk cases. So we can say that our model is good for medium to high risk cases.

Common Notes

- We have here directly used the LSTM model without exploring the other neural network models. LSTM is a memory based model which retains the long term behaviours to get the parameters updates for a time series analysis. LSTM is designed to handle the time series problems.

- LSTM models are data hungry architectures. We have bench-marked our model on rainfall prediction as it has higher number of samples as compared to malaria data and

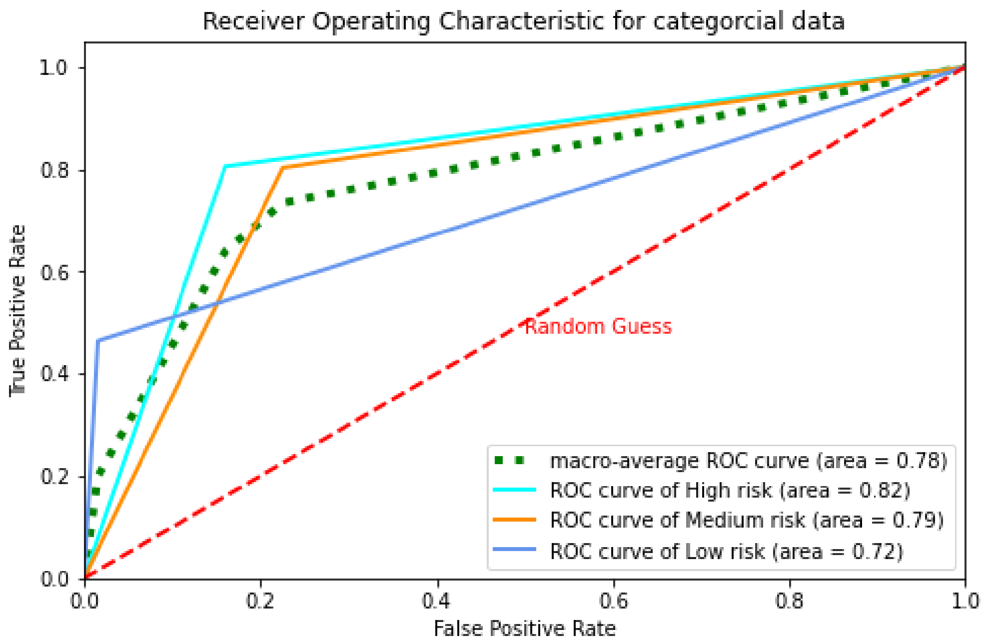


Figure 4.7: ROC score for temperature based malaria prediction for Patna district over 1 week lead time

then applied on the malaria prediction model.

- we have performed the prediction verification using the Heidke score and ROC curves for all the models but here we have shown only the best performing rainfall and malaria prediction models.
- The models are tested for the 38 different districts which have entirely different distribution and model have performed well on all the districts. We have also created a dashboard where use can put their own data to predict the malaria and rainfall values using the method of their choice

Chapter 5

Dashboard

The motivation behind creating this dashboard is to get an easy and rich visualisation on the go. The added cards with the dashboard contain the basic statistics about the variables. As mentioned, the dashboard was created in the python environment using Plotly dash. Plotly dash is a great library with very excellent documentation of functions and libraries.

Our dashboard has been deployed to the cloud on Heroku and can be accessed via the link below.

App Link: `https://imd-rishabh.herokuapp.com/`

This dashboard has multiple pages, and we will go through them one by one. Each page has an Info tab where the details about the operation are mentioned along with some technical details.

5.1 Page - 1: Let's Explore

USER: Select district and year range

As the name suggests, here will explore the data with data trends and basic statistics like mean, maximum and minimum for the data. Here the user can choose the district and range of year, and this page will display the same. By default, the range has been kept to 2 years but can be changed via the slider at the bottom. The cards on the page will visualise the basic statistics for each variable like Rainfall, Tmax, Tmin, Humidity and Malaria. The meteorological data has been kept at the same length as malaria data which is 8 years (2013 - 2020).

5.2 Page - 2: SOM Clusters

USER: Pick district of choice

This page explores the unsupervised clustering of variables using SOM. The first figure will show the clustering among the variables. This is our node point where we can observe the common pattern among the complex data. Next, we will have a bar plot to see the maximum value of each variable in the clusters. There is a total of 25 clusters in a 5x5 matrix. Users will pick the district

USER: Pick the cluster

We can visualise which weeks are clustered in the data and their correlation matrix based on cluster number.

5.3 Page - 3: Regression

USER: Pick the district and variable to be predicted (Malaria/Rainfall)

Here we will see the regression model performance over the 4 week lead time and skill scores for the lead one forecast. Patna, Nalanda, and Vaishali's results are pre-trained and saved for quick visualisation. Users can also set the district of choice and train and visualise the models on demand.

5.4 Page - 4: LSTM Model

USER:

- **Select District**
- **Select year range of data(default 40 years)**
- **Select variable (default Rainfall)**
- **Select target type (default: categorical) (options - series/categorical)**

Based on the user's choice, the model will run, train, forecast and visualise on demand. The results will show the learning curve and skill scores for up to 2 weeks of lead time and a skill score vs lead time plot for prediction validation. Also, we will get an actual vs predicted plot for the series and a confusion matrix for the categorical model. Pre-trained models for Patna and Nalanda have been saved for quick visualisation.

5.5 Page- 5: Malaria Prediction

USER:

- **Target - Series or categorical (Default - Series)**
- **District: Select district**

5.5.1 Tab - 1: Fourier Based LSTM Model

This is similar to the LSTM Model in the previous page, but it has been trained on the Fourier transformed data. The pre-trained models for Patna. Vaishali and Nalanda are stored for quick visualisation. The plots are similar to LSTM pages with learning curves, skill score and confusion matrices.

5.5.2 Tab - 2: Fourier Transformed Data

This tab will show the transformed malaria data on which the LSTM model has been trained. There will be a before and after data trend and peak frequencies obtained in the Fourier transform of noisy vs cleaned data.

5.6 Page - 6: Real time malaria forecast

User:

- **Rainfall: Latest rainfall value**
- **Tmax: Latest Tmax value**
- **Tmin: Latest Tmin value**
- **Humidity: Latest humidity value**
- **Malaria: latest malaria value**

This is a real-time application tab. Users can input the values of rainfall, tmax, tmin, humidity and malaria, we can see the live forecasting of malaria cases. We are using the previous 3 weeks value from the data set, which will be completely replaced by the 4 week user input in the newly launched dashboard for malaria prediction at IMD

5.7 Page - 7: Transmission Window Forecast

Here we will forecast the malaria values based on the window created by Tmax and Tmin predictions

USER:

- **Year: Select the year for which forecast is required**
- **District: Select District**

5.7.1 Temperature window based prediction

This tab will have prediction results for the test set, which is 2013 - 2020. So based on the year of choice, we will get a bar plot for week wise malaria severity predictions in terms of bars of a different colour for 4 week lead time in different subplots. We will have pre-trained results for all results.

5.7.2 Spatial Temperature Window based forecast

USER: Week - Choose the week from 2013 to 2020

This tab will see the malaria severity levels for all districts over the Bihar map for the week of choice. Different colour levels indicate the severity of malaria cases.

5.7.3 Transmission window based forecast model

This tab is similar to LSTM models and yields the same results too. We have pre-trained models for Patna, Nalanda and Vaishali. Based on the other district of choice, we can also live train and visualise.

We also developing another dashboard just for temperature based forecasts. This dashboard will be launched as a IMD product for malaria prevention in Bihar. We are also planned to replicate the models and predictions for the Odisha state.

Bibliography

[1] Who vector-borne diseases, . URL www.who.int/news-room/fact-sheets/detail/vector-borne-diseases.

[2] vbd-fact-sheets.pdf. https://www.who.int/docs/default-source/searo/india/health-topic-pdf/vbd-fact-sheets.pdf?sfvrsn=c1908b04_2#:~:text=Every%20year%20there%20are%20more,Japanese%20encephalitis%20and%20onchocerciasis%2C%20globally. (Accessed on 11/04/2021).

[3] Life cycle of aedes aegypti and ae. albopictus mosquitoes — mosquitoes — cdc. https://www.cdc.gov/mosquitoes/about/life-cycles/aedes.html. (Accessed on 10/28/2021).

[4] Céline Christiansen-Jucht, Paul E Parham, Adam Saddler, Jacob C Koella, and María-Gloria Basáñez. Temperature during larval development and adult maintenance influences the survival of anopheles gambiae s.s. 7(1), November 2014. doi: 10.1186/s13071-014-0489-3. URL https://doi.org/10.1186/s13071-014-0489-3.

[5] v54n3a21.indd. https://www.scielo.br/j/rbent/a/Twj8n8fN3S8tCbzVwHFHfvF/?lang=en&format=pdf. (Accessed on 10/28/2021).

[6] World malaria report 2020, . URL www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2020.

[7] PA PAHO. Epidemiological alert: increase in cases of malaria. *Pan American Health Organization. PAHO, Washington, DC*, 2017.

[8] World malaria report 2019, . URL https://www.who.int/docs/default-source/malaria/world-malaria-reports/9789240015791-double-page-view.pdf?sfvrsn=2c24349d_5.

[9] Bihar_-_disease_burden_profile[1].pdf. http://www.healthdata.org/sites/default/files/files/Bihar_-_Disease_Burden_Profile%5B1%5D.pdf. (Accessed on 10/27/2021).

[10] Ross Boyce, Raquel Reyes, Michael Matte, Moses Ntaro, Edgar Mulogo, Joshua Metlay, Lawrence Band, and Mark Siedner. Severe flooding and malaria transmission in the western ugandan highlands: Implications for disease control in an era of global climate change. *Journal of Infectious Diseases*, 214:1403–1410, 08 2016. doi: 10.1093/infdis/jiw363.

[11] National framework for malaria elimination — vikaspedia. https://vikaspedia.in/health/health-campaigns/national-framework-for-malaria-elimination. (Accessed on 10/27/2021).

[12] Aparup Das, Anupkumar R. Anvikar, Lauren J. Cator, Ramesh C. Dhiman, Alex Eapen, Neelima Mishra, Bhupinder N. Nagpal, Nutan Nanda, Kamaraju Raghavendra, Andrew F. Read, Surya K. Sharma, Om P. Singh, Vineeta Singh, Photini Sinnis, Harish C. Srivastava, Steven A. Sullivan, Patrick L. Sutton, Matthew B. Thomas, Jane M. Carlton, and Neena Valecha. Malaria in india: The center for the study of complex malaria in india. 121(3):267–273, March 2012. doi: 10.1016/j.actatropica.2011.11.008. URL https://doi.org/10.1016/j.actatropica.2011.11.008.

[13] Ashwani Kumar. Burden of malaria in india: Retrospective and prospective view. URL https://www.ncbi.nlm.nih.gov/books/NBK1720/.

[14] Prevalence of malaria and economic loss in two major iron ore mines in sundargarh district, orissa - pubmed. https://pubmed.ncbi.nlm.nih.gov/1810746/. (Accessed on 10/22/2021).

[15] Malaria: cost to india and future trends - pubmed. https://pubmed.ncbi.nlm.nih.gov/9031392/. (Accessed on 10/22/2021).

[16] D J Gubler, P Reiter, K L Ebi, W Yap, R Nasci, and J A Patz. Climate variability and change in the united states: potential impacts on vector- and rodent-borne diseases. 109(suppl 2):223–233, May 2001. doi: 10.1289/ehp.109-1240669. URL https://doi.org/10.1289/ehp.109-1240669.

[17] K. P. Paaijmans, A. F. Read, and M. B. Thomas. Understanding the link between malaria risk and climate. 106(33):13844–13849, August 2009. doi: 10.1073/pnas.0903423106. URL https://doi.org/10.1073/pnas.0903423106.

[18] Daniel J Weiss, Samir Bhatt, Bonnie Mappin, Thomas P Van Boeckel, David L Smith, Simon I Hay, and Peter W Gething. Air temperature suitability for plasmodium falciparum malaria transmission in africa 2000-2012: a high-resolution spatiotemporal prediction. 13(1), May 2014. doi: 10.1186/1475-2875-13-171. URL https://doi.org/10.1186/1475-2875-13-171.

[19] M.H. Craig, R.W. Snow, and D. le Sueur. A climate-based distribution model of malaria transmission in sub-saharan africa. *Parasitology Today*, 15(3):105–111, March 1999. doi: 10.1016/s0169-4758(99)01396-4. URL https://doi.org/10.1016/s0169-4758(99)01396-4.

[20] Torleif Markussen Lunde, Mohamed Nabie Bayoh, and Bernt Lindtjørn. How malaria models relate temperature to malaria transmission. *Parasites & Vectors*, 6 (1), January 2013. doi: 10.1186/1756-3305-6-20. URL https://doi.org/10.1186/1756-3305-6-20.

[21] Maquins Odhiambo Sewe, Yesim Tozan, Clas Ahlm, and Joacim Rocklöv. Using remote sensing environmental data to forecast malaria incidence at a rural district hospital in western kenya. 7(1), June 2017. doi: 10.1038/s41598-017-02560-z. URL https://doi.org/10.1038/s41598-017-02560-z.

[22] Forecasting malaria cases using climatic factors in delhi, india: A time series analysis. https://www.hindawi.com/journals/mrt/2014/482851/. (Accessed on 10/22/2021).

[23] Ipcc — intergovernmental panel on climate change. https://www.ipcc.ch/. (Accessed on 10/28/2021).

[24] P Reiter. Climate change and mosquito-borne disease. 109(suppl 1):141–161, March 2001. doi: 10.1289/ehp.01109s1141. URL https://doi.org/10.1289/ehp.01109s1141.

[25] T. H. Jetten, W.J.M. Martens, and W. Takken. Model simulations to estimate malaria risk under climate change. 33(3):361–371, May 1996. doi: 10.1093/jmedent/33.3.361. URL https://doi.org/10.1093/jmedent/33.3.361.

[26] Kate Zinszer, Ruth Kigozi, Katia Charland, Grant Dorsey, Timothy F Brewer, John S Brownstein, Moses R Kamya, and David L Buckeridge. Forecasting malaria in a highly endemic country using environmental and clinical predictors. 14(1), June 2015. doi: 10.1186/s12936-015-0758-4. URL `https://doi.org/10.1186/s12936-015-0758-4`.

[27] Odu Nkiruka, Rajesh Prasad, and Onime Clement. Prediction of malaria incidence using climate variability and machine learning. 22:100508, 2021. doi: 10.1016/j.imu.2020.100508. URL `https://doi.org/10.1016/j.imu.2020.100508`.

[28] Mohammad Y. Anwar, Joseph A. Lewnard, Sunil Parikh, and Virginia E. Pitzer. Time series analysis of malaria in afghanistan: using ARIMA models to predict future trends in incidence. 15(1), November 2016. doi: 10.1186/s12936-016-1602-1. URL `https://doi.org/10.1186/s12936-016-1602-1`.

[29] Hamid H. Hussien, Fathy H. Eissa, and Khidir E. Awadalla. Statistical methods for predicting malaria incidences using data from sudan. 2017:1–9, March 2017. doi: 10.1155/2017/4205957. URL `https://doi.org/10.1155/2017/4205957`.

[30] Neural networks and the universal approximation theorem — by milind sahay — towards data science. `https://towardsdatascience.com/neural-networks-and-the-universal-approximation-theorem-8a389a33d30a`. (Accessed on 10/25/2021).

[31] Cyril Caminade, Sari Kovats, Joacim Rocklov, Adrian M Tompkins, Andrew P Morse, Felipe J Colón-González, Hans Stenlund, Pim Martens, and Simon J Lloyd. Impact of climate change on global malaria distribution. *Proceedings of the National Academy of Sciences*, 111(9):3286–3291, 2014.

[32] The self-organizing map - proceedings of the ieee. `https://sci2s.ugr.es/keel/pdf/algorithm/articulo/1990-Kohonen-PIEEE.pdf`. (Accessed on 10/23/2021).

[33] Self-organizing maps. `http://gorayni.github.io/blog/2014/10/08/som.html`. (Accessed on 10/23/2021).

[34] Felix M. Riese. SuSi: SUpervised Self-organIzing maps in Python. doi.org/10.5281/zenodo.2609130, 2019.

[35] Felix M. Riese, Sina Keller, and Stefan Hinz. Supervised and Semi-Supervised Self-Organizing Maps for Regression and Classification Focusing on Hyperspectral Data.

Remote Sensing, 12(1), 2020. ISSN 2072-4292. doi: 10.3390/rs12010007. URL https://www.mdpi.com/2072-4292/12/1/7.

[36] V Moosavi, S Packmann, and I Vallés. Sompy: A python library for self organizing map (som), 2014. GitHub.[Online]. Available: https://github. com/sevamoo/SOMPY.

[37] [1702.07800] on the origin of deep learning. https://arxiv.org/abs/1702.07800. (Accessed on 11/08/2021).

[38] Gentle introduction to echo state networks — by madalina ciortan — towards data science. https://towardsdatascience.com/gentle-introduction-to-echo-state-networks-af99e5373c68. (Accessed on 10/23/2021).

[39] Michael E Loevinsohn. Climatic warming and increased malaria incidence in rwanda. *The Lancet*, 343(8899):714–718, 1994.