TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

THESIS NO: 072/MSCSKE/668

Recurrent Neural Network Based Forecasting of Crop Production in Nepal

by
Surendra Joshi

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
LALITPUR, NEPAL

NOVEMBER, 2018

Recurrent Neural Network Based Forecasting of Crop Production in Nepal

by

Surendra Joshi

072/MSCSKE/668

<u>Thesis Supervisor:</u>

Dr. Aman Shakya

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer System and Knowledge Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

November, 2018

COPYRIGHT

The author has agreed that the library, Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, may make this thesis freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this thesis work for scholarly purpose may be granted by the professor(s), who supervised the thesis work recorded herein or, in their absence, by the Head of the Department, wherein this thesis was done. It is understood that the recognition will be given to the author of this thesis and to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering in any use of the material of this thesis. Copying of publication or other use of this thesis for financial gain without approval of the Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus and author's written permission is prohibited.

Request for permission to copy or to make any use of the material in this thesis in whole or part should be addressed to:

Head
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering
Pulchowk, Lalitpur
Nepal

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

APPROVAL PAGE

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled "**Recurrent Neural Network Based Forecasting of Crop Production in Nepal**", submitted by Surendra Joshi in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Computer Systems and Knowledge Engineering**".

………………………………………...

Supervisor: Dr. Aman Shakya
Program Coordinator
Department of Electronics and Computer Engineering

………………………………………...

External Examiner: Dr. Roshan Koju
Nepal Telecom

………………………………………...

Committee Chairperson: Dr. Aman Shakya
Program Coordinator
Department of Electronics and Computer Engineering

………………………………………...

Date

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Recurrent Neural Network Based Forecasting of Crop Production in Nepal",** submitted by **Surendra Joshi** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Computer System and Knowledge Engineering**" has been accepted as a bonafide record of work independently carried out by him in the department.

Dr. Surendra Shrestha
Head of the Department
Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University,
Nepal.

ACKNOWLEDGEMENT

A lot of people deserve a great deal of thanks for their tremendous help and support in completing this study. I would like to express my sincere thanks to my respected Supervisor cum Program Coordinator of Department of Electronics and Computer Engineering, Pulchowk Campus, **Dr. Aman Shakya**, for regularly guiding and granting his precious time to provide feedback, comments and continuous support in the research.

I would further take the opportunity to convey my gratitude to Head of Department **Dr. Surendra Shrestha.** Furthermore, I would like to thank **Prof. Dr. Shashidhar Ram Joshi**, **Prof. Dr. Subarna Shakya**, **Dr. Dibakar Raj Pant**, **Dr. Sanjeeb P. Panday** and **Dr. Roshan Koju** for their valuable comments and guidance during the study.

I would also like to express my gratitude to Agricultural experts, **Dr. Bhoj Raj Joshi** and **Dr. Tek Bahadur Bam** for providing valuable insights and continuous support to prepare, analyze and confirm the accuracy of the model. Without their support, this study would have been impossible to accomplish

I would like to thank Ministry of Agriculture and Livestock Development, Department of Hydrology and Meteorology, Central Bureau of Statistics, Department of Food Technology and Quality Control, Nepal Agriculture Input Supply Company Ltd. and Center for Environmental and Agricultural Policy Research, Extension and Development for providing all the data required for thesis study. I would also like to thank Department of Electronics and Computer Engineering, Pulchowk Campus, for providing necessary resources and information.

The suggestions and guidance of all the people who have helped me are much appreciated.

ABSTRACT

In this study, Gated Recurrent Unit (GRU) model was used for predicting Rice crop production in Nepal using climatic and fertilizer variables. The climatic variables used were Maximum Temperature, Minimum Temperature, Morning Humidity, Evening Humidity and Rainfall and Ecological Regions and fertilizer variables were Nitrogen, Phosphorous, Potassium and Compost. When the model was trained on 70% of data and tested on 30% of the data, the accuracy of the model was 81% for predicting the production. When tested on year 2016, accuracy of the model was 81.33% and for year 2017, the accuracy of the model was 73.33%. While GRU was compared with baseline Artificial Neural Network (ANN) with same architecture for Siraha district, it performed better than baseline ANN. But when input variables were increased, it performed even better. This proved that GRU can be used for optimal prediction of Rice crop in Nepal.

Keywords: ANN, Climate, Fertilizer, GRU, RNN

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
ARIMA	:	Autoregressive Integrated Moving Average
CBS	:	Central Bureau of Statistics
DHM	:	Department of Hydrology and Meteorology
DLM	:	Dynamic Linear Models
ES	:	Exponential Smoothing
FN	:	False Negatives
FP	:	False Positives
GDP	:	Gross Domestic Product
GoN	:	Government of Nepal
GRU	:	Gated Recurrent Unit
Ha	:	Hectares
MLFANN	:	Multi-Layered Feed-forward Artificial Neural Network
MLR	:	Multiple Linear Regression
MOAD	:	Ministry of Agriculture Development
MoF	:	Ministry of Finance
MT	:	Metric Tonnes
PPV	:	Positive Predicted Value
RH	:	Relative Humidity
RMSE	:	Root Mean Square Error
RNN	:	Recurrent Neural Network
TN	:	True Negatives
TP	:	True Positives

CHAPTER I: INTRODUCTION

1.1. Background

Agriculture is the main occupation of 65% of Nepalese people and it contributes to one third of the National Gross Domestic Product[1] . Rice, Maize, Wheat and Potato are the major crops in terms of area coverage, production and food supply. Total agricultural land is about 2 million hectare (ha) where Rice, Maize, Wheat and Potato are grown as number one, two, three and four crops by area respectively. Rice is the number one crop among the major crops in terms of area coverage (65%), production and food supply.

The population growth rate is 2.1% but the agriculture growth rate is not meeting the increased food requirement of the country. So, import of agriculture products is increasing every year[1].

Nepal is divided into three ecological regions: Terai, Hill and Mountain. The Mountain Region is situated above 3000 meters in the north, the Hill Region is situated between 500 to 2000 meters in the middle, and the Terai Region is situated below 500 meters in the south of Nepal. Each of these regions have a distinct climatic and geographical setting [1]. Based on the climatic condition, there are three seasons: summer which falls on March to June, rainy (monsoon) season is July to October and winter is November to February. Approximately 80% of rainfall occurs during monsoon periods. There is less rain in far-western and mid-western regions, moderate rain in western region and heavy rain in the eastern region. The varying temperature and rainfall directly affects the production of crops.

Despite of the agriculture being main occupation of people and major contributor to GDP, there is no defined and structured methodology and tools to calculate the food requirement, and food deficit at local and national level. Agricultural production plans are prepared without consideration of food requirement which creates uncertainty about import of food items. Similarly, area to be cultivated under each food crop

[1] Ministry of Finance (2017). Budget Speech of Fiscal Year 2017/18

remains undefined thereby creating uncertainty about planning of food crop within specific time period.

In Nepal, among all the food crops, Rice is consumed as preferred food item by almost all the population of Nepal. But the production of rice does not match with the demand. Therefore, a solution is required to predict whether there is deficit or surplus of rice crop production in Nepal. Considering this fact, this study focuses on using Gated Recurrent Unit (GRU), improved version of Recurrent Neural Network (RNN) for optimal prediction of Rice crop production in Nepal using agricultural data. RNN is a type of neural network that has an internal loop. Recurrent networks include a feedback loop, whereby output from step n-1 is fed back to the network to affect the outcome of step n and so forth for each subsequent step.

1.2. Statement of the Problem

- In Nepal, Agricultural production and import plan is done on ad-hoc basis without considering the food need of the people living in specific geographic regions of the country resulting deficit in some places while surplus in other areas.
- Traditional approaches estimate yields by calibrating regression models with predictive variables, spectral information from remote sensing data, or a combination of them. The temporal characteristics of predictive variables are not fully leveraged and often treated as independent observation in model inputs without accounting for the potential accumulative effects.
- The limitation of using Artificial Neural Network (ANN) for prediction of production is that they have no memory and input shown to them is processed independently, with no state kept in between inputs.
- Similarly, researches conducted so far on crop yield prediction are done either considering the whole country as one ecological region or it is done on same ecological regions only.
- This study focuses on forecasting Rice crop production in Nepal applying GRU model using fertilizer and climate variables. This study has used agricultural data from all 75 districts of Nepal for 26 years considering the different ecological variables that directly affect production. GRU model has its own internal memory. It consists of an update gate and a reset gate. The update gate defines how much

previous memory to keep around and the reset gate defines how to combine the new input with the previous memory to fully utilize sequential and temporal characteristics of the predictive variables for forecasting crop production.

1.3. Objectives

The objectives of this study are:

- To forecast Rice crop production deficit or surplus in Nepal using GRU model using past agricultural data
- To compare the accuracy of GRU model with the baseline ANN model

1.4. Scope of the Work

This study is focused on creating GRU model for accurately forecasting the production of Rice crop in Nepal considering different ecological variables that differ in each district and directly affect the crop production. This model is expected to be useful for Ministry of Agricultural and Livestock Development, Non-governmental Organizations and Private Sector agencies working in the field of agriculture, and Agricultural Experts and Statisticians for estimating Food Surplus or Deficit of Rice crop in Nepal. On the basis of surplus/deficit of Rice production forecasted by this model, government and concerned organizations can develop rice production plan, rice import plan and rice distribution plan for different parts of the country.

This study will also serve as a reference material and provide guidance to future researchers in this area in the context of Nepal.

1.5. Thesis Structure

This thesis report is structured into five chapters. Chapter 1 contains General Background about the agriculture in Nepal. Chapter 2 consists of overview on the Literature Reviews and describes what works and researches have been conducted in the past. Chapter 3 contains methodology that provides a detailed explanation of the GRU model, its' network structure and hyper-parameters that has been used for training GRU model. Chapter 4 contains Results and Discussion of the GRU model. Chapter 5 contains the Conclusion and Future works.

CHAPTER II: LITERATURE REVIEW

The review of empirical works associated with prediction on agriculture production using different methods on different periods is presented in this section. Review of different works are divided according to the methods used for forecasting as shown below:

2.1. Agricultural Prediction Using Statistical Forecast Models

Shashtri et. al. [2] used regression model for the prediction of Maize, Wheat and Cotton yield in India. This study used quadratic, pure quadratic, linear, polynomial, generalized linear regression and stepwise linear regression models. The accuracy of results obtained from them were compared using R^2, Root Mean Square Error (RMSE) and Mean Percentage Prediction Error (MPPE). The study concluded that Generalized Linear Model has lower RMSE value than other models for Wheat Yield Prediction, Pure Quadratic Model has lower RMSE value than other models for Maize Yield Prediction and proposed Stepwise Linear Regression Model has lower RMSE values than the other models.

Sellam & Poovmmal [3] analyzed the environmental parameters like Area under Cultivation, Annual Rainfall and Food Price Index for a period of 10 years from 1990-2000 that influence the yield of crop and to establish a relationship among the parameters. In this research, Linear Regression (LR) was used to establish relationship between explanatory variables (Area under Cultivation, Annual Rainfall and Food Price Index) and the crop yield as response variable. R^2 value showed that yield is mainly dependent on Annual Rainfall Area under Cultivation and Food Price Index are the other two factors influencing the crop yield.

Sahu et. al. [4] used Jenkins Autoregressive Integrated Moving Average (ARIMA) modelling technique to predict area, production, yield and total seed of rice (paddy) and wheat for seven SAARC countries, except Maldives. The forecast showed that rice and wheat production for the year 2020 would be about 794 and 777 million tons respectively in the world. In-spite of increase in production the study revealed that the yield of rice and wheat in world would be 4.35 t/ha and 3.4 t/ ha in 2020 but the yield of these two crops in SAARC countries, barring one country in each, will remain far below the world projection. Thus, under the given remote possibility of horizontal expansion, the study emphasized the need for quantum jump in the per hectare yield of these two crops for this region. The study advocated that good quality of seeds in good amount be made available to the farmers, otherwise the whole food security of this part of the Globe would be under tremendous risk.

Bhatti et. al. [5]compared three methods: Box-Jenkins' ARIMA, Dynamic Linear Models (DLM) and exponential smoothing (ES) to forecast future crop production levels using time series data for four major crops in Pakistan: wheat, rice, cotton and pulses. The various measures of forecast accuracy, namely the root mean squared error (RMSE), mean absolute percentage error (MAPE) and mean absolute scaled error (MASE) were also calculated for all data sets over the forecast period for each model. According to these accuracy measures, the exponential smoothing method came out as the best for wheat and rice time series, and second-best for cotton and pulses time series. The DLM performed best for the cotton time series, while the Box-Jenkins ARIMA technique was best for the pulses time series.

2.2. Agricultural Prediction using ANN models

Ranjeet & Armstrong [6] used Artificial Neural Networks (ANNs) using back propagation for predicting paddy crop cultivation in Siraha district of Nepal using agricultural data from thirteen years. Climatic parameters including rainfall, maximum temperature and minimum temperature along with the fertilizers - Urea, Dap and Potash were used as input values. The experiment showed that the trained neural network produced a minimum sum of squared error of 1.471 and relative error of 0.302 which indicated that the test model is capable of predicting crops yield in Nepal.

Lamba & Dhaka [7] conducted study on Wheat Yield Prediction Using Artificial Neural Network and Crop Prediction Techniques. The paper represented the forecasting techniques in Wheat crop. The major forecasting models were Statistical, Metrological, Simulation, Agronomic, Remote Satellite Sensed, Synthetic and Mathematical in the field of Agriculture. This paper presented compact combination of all these models and showed why Neural Network Model is important from other models for nonlinear data behavior system like wheat crop yield prediction.

Rode & Dahikar [8] conducted study on Agricultural Crop Yield Prediction Using Artificial Neural Network Approach. ANN has been used to predict the suitable crop among Bajara, Soyabean, Corn, Wheat, Rice and Groundnut in India by sensing various parameter of soil and also parameter related to atmosphere. Parameters like type of soil, PH, nitrogen, phosphate, potassium, organic carbon, calcium, magnesium, sulphur, manganese, copper, iron, depth, temperature, rainfall, humidity. This paper showed the ability of artificial neural network technology to be used for the approximation and prediction of crop yields at rural district.

Stansy et. al. [9] focused on prediction of crop yield levels, using an artificial intelligence approach, namely a multi-layer neural network model. The study implemented multi-layer neural network for the prediction of the Onion crop yield, using Density of nursling per meter square and compared the accuracy of this approach with the accuracy of the well-known regression model designed for the prediction of empirical data using Residual Sum of Squares method (RSS). The use of a multi-layer neural network proved to be more accurate than regressive model.

Singh and Prajneshu [10] used ANN Multilayered Feed Forward Artificial Neural Network (MLFANN) for Modelling and Forecasting Maize Crop Yield in India. To train such a network, two types of learning algorithms, namely Gradient descent algorithm (GDA) and Conjugate gradient descent algorithm (CGDA) were discussed. The methodology was illustrated by considering maize crop yield data as response variable and total human labor, farm power, fertilizer consumption, and pesticide consumption as predictors. They found that a three-layered MLFANN with (11,16) units in the two hidden layers performed best in terms of having minimum mean square errors (MSE) for training, validation and test sets. Superiority of this MLFANN over Multiple Linear Regression (MLR) analysis has also been demonstrated for the maize data considered in the study.

Ji et. al. [11] developed an agricultural management system to predict rice yields in the planning process. Field-specific rainfall data and the weather variables (daily sunshine hours, daily solar radiation, daily temperature sum and daily wind speed) were used for each location. The necessity of the study were to identify whether artificial neural network (ANN) models could effectively predict rice yield for typical climatic conditions of the mountainous region, evaluate ANN model performance relative to variations of developmental parameters and compare the effectiveness of multiple linear regression models with ANN models. Optimal learning rates were between 0.71 and 0.90. ANN models consistently produced more accurate yield predictions than regression models. ANN rice grain yield models for Fujian resulted in R^2 and RMSE of 0.67 and 891 versus 0.52 and 1977 for linear regression, respectively. Although more time consuming to develop than multiple linear regression models, ANN models proved to be superior for accurately predicting rice yields under typical Fujian climatic conditions.

Table 2.1: Comparative Chart of Different Literature Reviews

Name	Method Used	Predicting Variable	Conclusion
Shashtri et. al. [2]	Quadratic, Pure Quadratic, Linear, Polynomial, Generalized Linear Regression and Stepwise Linear Regression	Wheat, Maize and Cotton Yield	Proposed regression model can be used for yield prediction in India
Sellam & Poovmmal [3]	Linear Regression	Rice crop yield	Rice yield is mainly dependent on Annual Rainfall
Sahu et. al. [4]	ARIMA	Area, Yield and Total seed Production of Rice and wheat	Per hectare yield and quality of seeds supplied to the farmers should be increased drastically
Bhatti et. al. [5]	ARIMA, DLM and ES	Wheat, Rice, Cotton and Pulse Production	ES was best for wheat and Rice crops, DLM was best for cotton crop and ARIMA was best for pulse crop.
Ranjeet & Armstrong [6]	ANN	Rice crop yield	ANN is capable of predicting Rice crop

			yield in Nepal
Lamba & Dhaka [7]	ANN	Wheat crop yield	ANN is very efficient compared to other statistical forecast models in agriculture for non-linear data
Rode & Dahikar [8]	ANN	Select Suitable crop among Bajara, Soyabean, Corn, Wheat, Rice and Groundnut	ANN is beneficial tool for predicting suitable crop
Stansy et. al. [9]	ANN and Regressive model	Onion Crop Yield	Multi-Layer ANN proved to be more accurate than Regressive model for Onion crop yield prediction.
Singh and Prajneshu [10]	ANN and MLR model	Maize Crop Yield	MLFANN is superior over MLR for predicting Maize crop yield
Ji et. al. [11]	ANN and MLR model	Rice crop yield	Although more time consuming than MLR models, ANN models proved to be superior for accurately predicting rice yields under typical Fujian climatic conditions.

CHAPTER III: METHODOLOGY

3.1. Overview of the proposed method

The overview of the methodology used in this study is presented in Figure 3.1. All the steps are explained in section 3.2.

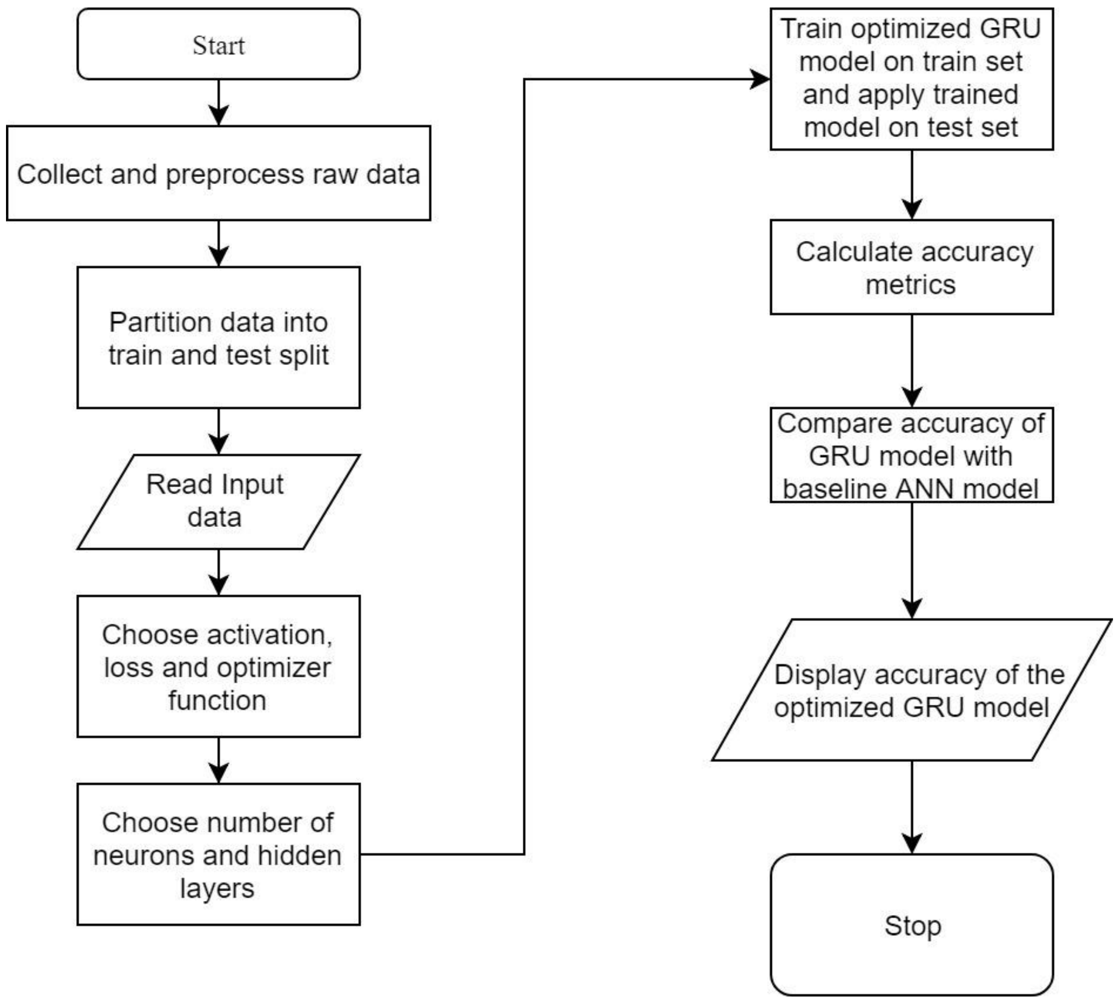


Figure 3.1: Flow-diagram of the study

3.2. Data Collection and Preprocessing

3.2.1. Description of Population and Sample

The population of this research includes main crop grown in Nepal. There are altogether three categories of crops: food crop, cash crop and industrial crop. Among these three categories of crops, food crop is used for the study purpose.

Nepal has been growing six types of food crops. They are Rice, Maize, Wheat, Millet, Barley and Potato. For this study, Rice crop has been used. Further, the study used the samples of 75 districts for 26 years starting from 1991 to 2016 as shown in the Table 3.1.

Table 3.1: Input data, total sample and their sources

S.N.	Variables Used	Total Sample	Duration	Reference	Source
1	Population	1950	1991-2016	National Population and Housing Census, 2017	Central Bureau of Statistics (CBS), Government of Nepal (GON)
2	Land Area	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development (MOAD), Government of Nepal
3	Food Production (MT)	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development, Government of Nepal
4	Nitrogen (MT)	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development, Government of Nepal
5	Potassium (MT)	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development, Government of Nepal
6	Compost (MT)	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development, Government of Nepal
7	Phosphorous (MT)	1950	1991-2016	Statistical Information on Nepalese Agriculture, 2017	Ministry of Agriculture Development , Government of Nepal
8	Maximum Temperature (^{O}C)	1950	1991-2016	Statistical Information on Weather Data of Nepal,2017	Department of Hydrology and Meteorology(DHM), Government of Nepal
9	Minimum Temperature (^{O}C)	1950	1991-2016	Statistical Information on Weather Data of Nepal,2017	Department of Hydrology and Meteorology(DHM), Government of Nepal
10	Relative Humidity	1950	1991-2016	Statistical Information on Weather Data of Nepal,2017	Department of Hydrology and

	Morning (%)					Meteorology, Government of Nepal
11	Relative Humidity Evening (%)		1950	1991-2016	Statistical Information on Weather Data of Nepal,2017	Department of Hydrology and Meteorology, Government of Nepal
12	Rainfall(mm)		1950	1991-2016	Statistical Information on Weather Data of Nepal,2017	Department of Hydrology and Meteorology, Government of Nepal
13	Mountain Region	18 districts	468	1991-2016	National Population and Housing Census, 2017	Central Bureau of Statistics, Government of Nepal
	Hill Region	37 districts	962	1991-2016		
	Terai Region	20 districts	520	1991-2016		
13	Food Requirement Per Head				Food Compository Table, Nepal, 2017	Department of Food Technology and Quality Control(DFTQC), Government of Nepal

Table 3.1 shows the total data that are used in this study. Population Census Data were collected from CBS, GoN [12]. Food Production, Land Area, Fertilizer data Nitrogen, Potassium, Phosphorous and Compost were collected from MOAD, GON [13]. Climatic Parameters Maximum Temperature, Minimum Temperature, RH (morning), RH (evening) and Rainfall were collected from DHM, GoN [14]. Food requirement per head data was collected from DFTQC, GoN [15].

3.2.2. Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format for RNN. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. If important data inputs are missing, then the effect on the neural network's performance can be significant. Developing a workable neural network application can be considerably more difficult without a solid understanding of the problem domain. Data pre-processing steps used in this study are:

a) Data Integration

The data from different Governmental agencies were collected and compiled into a single file.

b) Data Cleaning

Due to the wide range of sources of information, information may be incomplete and noisy. Therefore, the data are cleaned to ensure the integrity and accuracy of the information. For yearly data, since Rice is grown from month June to October, climatic parameters such as Rainfall, is the cumulative value of a total of five months measured in mm. Similarly, Maximum and minimum temperature and Relative Humidity are averaged data measured from June to October.

c) Data Rescaling

The given data was first rescaled. The fertilizer data Nitrogen, Potassium, Phosphorous and Compost data was rescaled from MT to MT/Ha by dividing with the total Crop Area of that district.

$$\text{Rescaled data(MT/Ha)} = \frac{\text{Data (MT)}}{\text{Crop Area (Ha)}} \quad \text{…(equation 3.2.1)}$$

(source: DOFTQC, 2017. See Table 3.1 for details)

d) Food Requirement Calculation

Food Requirement per head has been predefined by Department of Food Technology and Quality Control, Government of Nepal. It is shown in the Table 3.2:

Table 3.2: Food Requirement per head

Crop	Food requirement per head per year (kg/head)
Rice	161.2

(source: DOFTQC, 2017. See Table 3.1 for details)

Table 3.2 shows the food requirement per head per year. The food requirement per head per year data was converted to MT and multiplied by total Population of that district in that year to give required production of Rice of that particular district.

$$\text{Required Production(MT)} = \frac{\text{kg}}{1000} * \text{Total population of that area} \quad \text{...(equation 3.2.2)}$$

(source: DOFTQC, 2017. See Table 3.1 for details)

Then the percentage difference between Actual Production and Required production was calculated as shown below:

$$\text{Percentage Difference(x\%)} = \frac{\text{Actual Production} - \text{Required Production}}{\text{Required Production}} * 100 \quad \text{…(equation 3.2.3)}$$

(source: DOFTQC, 2017. See Table 3.1 for details)

The snapshot of data after rescaling and food requirement calculation is shown in Table 3.3.

Table 3.3: Sample of manually integrated agricultural data

S.N.	Districts	Region	Population	Area1991 (Ha)	N MT/ha	P MT/ha	K MT/ha	Compost MT/ha	Maximum Temperature (0C)	Minimum Temperature (0C)	Relative Humidity (%) (8:45 am)	Relative Humidity (%) (5:45 pm)	Rainfall (mm)	Actual Production (MT)	Required Production (MT)
1	Achham1991	H	198188	5290	0.019607	0.001493	0.000008	0.752108	18.940000	11.980000	62.240000	44.800000	449.600000	10900	31947.905600
2	Arghakhanchi	H	180884	4820	0.019090	0.001454	0.000008	0.732300	22.520000	16.580000	90.700000	92.620000	437.758828	9670	29158.500800
3	Baglung	H	232488	5190	0.022111	0.001684	0.000009	0.848183	31.640000	20.900000	79.420000	76.920000	507.032067	12060	37477.065600
4	Baitadi	H	200716	5000	0.017128	0.001304	0.000007	0.657025	27.480000	18.620000	79.460000	74.600000	392.760661	9000	32355.419200
5	Bajhang	M	139092	3800	0.006069	0.000462	0.000003	2.048665	29.100000	17.320000	73.660000	62.620000	1440.030933	7880	22421.630400
6	Bajura	M	92010	2630	0.005575	0.000425	0.000002	1.881960	27.040000	8.800000	83.640000	80.300000	486.610647	5010	14832.012000
7	Banke	T	187400	32250	0.021608	0.001645	0.000009	0.142096	33.940000	22.820000	84.560000	76.840000	436.400734	81400	30208.880000
8	Bara	T	415718	60220	0.025811	0.001965	0.000011	0.169733	32.540000	24.060000	81.840000	72.160000	392.760661	181560	67013.741600
9	Bardiya	T	285604	32150	0.021409	0.001630	0.000009	0.140786	33.700000	24.360000	84.040000	79.360000	499.029236	80400	46039.364800
10	Bhaktapur	H	172952	4770	0.062799	0.004782	0.000026	2.408939	28.220000	18.580000	80.180000	82.580000	525.568019	31480	27879.862400
11	Bhojpur	H	141903	12820	0.021221	0.001616	0.000009	0.814021	24.720000	17.280000	88.360000	85.140000	373.177745	28590	22874.763600
12	Chitwan	T	354488	28000	0.028221	0.002149	0.000012	0.185579	33.220000	24.040000	84.620000	79.100000	456.767831	92300	57143.465600
13	Dadeldhura	H	104647	6500	0.019031	0.001449	0.000008	0.730028	23.060000	16.160000	85.120000	75.740000	404.989933	13000	16869.096400
14	Dailekh	H	225768	5000	0.017128	0.001304	0.000007	0.657025	26.960000	17.960000	84.740000	82.400000	440.899711	9000	36393.801600
15	Dang Deokhu	T	354413	36550	0.021697	0.001652	0.000009	0.142676	29.940000	21.560000	82.360000	73.580000	389.909178	92630	57131.375600
16	Darchula	M	101683	1650	0.005499	0.000419	0.000002	1.856120	31.680000	19.800000	90.440000	72.300000	498.087090	3100	16391.299600
17	Dhading	H	278068	13100	0.021762	0.001657	0.000009	0.834795	28.740000	20.620000	85.240000	81.560000	1043.055467	29960	44824.561600
18	Dhankuta	H	146386	9250	0.022920	0.001745	0.000009	0.879190	25.040000	18.220000	82.600000	88.420000	921.814520	22280	23597.423200
19	Dhanusha	T	543672	53580	0.019552	0.001489	0.000008	0.128575	32.560000	25.200000	80.600000	73.600000	386.935090	122370	87639.926400
20	Dolakha	M	173236	2110	0.007018	0.000534	0.000003	2.369171	23.620000	15.140000	83.620000	88.000000	1025.541725	5060	27925.643200
21	Dolpa	M	25013	520	0.004390	0.000334	0.000002	1.481902	20.400000	8.300000	74.780000	80.700000	448.028397	780	4032.095600
22	Doti	H	167168	6730	0.016274	0.001239	0.000007	0.624266	34.580000	20.920000	79.060000	60.280000	571.870305	11510	26947.481600
23	Gorkha	H	252524	16820	0.019919	0.001517	0.000008	0.764099	28.600000	20.140000	87.600000	81.620000	480.722683	35210	40706.868800

(source: CBS, MOAD, DHM,DFTQC,2017. See table 3.1 for details)

Table 3.3 shows sample of manually integrated agricultural data. Here, H indicates Hill Region, M indicates Mountain Region and T indicates Terai Region. Ha means Hectares, N means Nitrogen fertilizer, P means Phosphorous fertilizer and K means Potassium fertilizer.

e) Data Classification

Amongst all the above variables shown in table 3.3, 10 input variables namely, Region, Nitrogen, Potassium, Phosphorous, Compost, Maximum Temperature, Minimum Temperature, Relative Humidity (Morning), Relative Humidity (Evening), Rainfall and one output variable Production was used for Neural Network .

The output Production data were classified into three classes according to the percentage difference as suggested by Agricultural Experts. For distributing data into three classes, following cases were used:

- **i.** If $\mathbf{x} > 10\%$, Classify as Surplus
- **ii.** If $\mathbf{x} < 10\%$, Classify as Deficit
- **iii.** If $\mathbf{-10 \leq x\% \leq 10}$, Classify as Normal

The snapshot of data after classification is shown in Table 3.4. More samples of data is shown in Annex A:

Table 3.4: Sample of data after classification of output data

Region	N MT/ha	P MT/ha	K MT/ha	Compost MT/ha	Maximum Temperature (0C)	Minimum Temperature (0C)	Relative Humidity (%) (8:45 am)	Relative Humidity (%) (5:45 pm)	Rainfall (mm)	Production
H	0.019607	0.001493	0.000008	0.752108	18.940000	11.980000	62.240000	44.800000	449.600000	DEFICIT
H	0.019090	0.001454	0.000008	0.732300	22.520000	16.580000	90.700000	92.620000	437.758828	DEFICIT
H	0.022111	0.001684	0.000009	0.848183	31.640000	20.900000	79.420000	76.920000	507.032067	DEFICIT
H	0.017128	0.001304	0.000007	0.657025	27.480000	18.620000	79.460000	74.600000	392.760661	DEFICIT
M	0.006069	0.000462	0.000003	2.048665	29.100000	17.320000	73.660000	62.620000	1254.400000	DEFICIT
M	0.005575	0.000425	0.000002	1.881960	27.040000	8.800000	83.640000	80.300000	1152.326108	DEFICIT
T	0.021608	0.001645	0.000009	0.142096	33.940000	22.820000	84.560000	76.840000	959.300000	SURPLUS
T	0.025811	0.001965	0.000011	0.169733	32.540000	24.060000	81.840000	72.160000	1145.880111	SURPLUS
T	0.021409	0.001630	0.000009	0.140786	33.700000	24.360000	84.040000	79.360000	950.462157	SURPLUS
H	0.062799	0.004782	0.000026	2.408939	28.220000	18.580000	80.180000	82.580000	1440.030933	SURPLUS
H	0.021221	0.001616	0.000009	0.814021	24.720000	17.280000	88.360000	85.140000	486.610647	SURPLUS
T	0.028221	0.002149	0.000012	0.185579	33.220000	24.040000	84.620000	79.100000	1252.862552	SURPLUS
H	0.019031	0.001449	0.000008	0.730028	23.060000	16.160000	85.120000	75.740000	436.400734	DEFICIT
H	0.017128	0.001304	0.000007	0.657025	26.960000	17.960000	84.740000	82.400000	392.760661	DEFICIT
T	0.021697	0.001652	0.000009	0.142676	29.940000	21.560000	82.360000	73.580000	963.216783	SURPLUS
M	0.005499	0.000419	0.000002	1.856120	31.680000	19.800000	90.440000	72.300000	1136.504538	DEFICIT
H	0.021762	0.001657	0.000009	0.834795	28.740000	20.620000	85.240000	81.560000	499.029236	DEFICIT
H	0.022920	0.001745	0.000009	0.879190	25.040000	18.220000	82.600000	88.420000	525.568019	NORMAL
T	0.019552	0.001489	0.000008	0.128575	32.560000	25.200000	80.600000	73.600000	868.024552	SURPLUS
M	0.007018	0.000534	0.000003	2.369171	23.620000	15.140000	83.620000	88.000000	1450.646138	DEFICIT
M	0.004390	0.000334	0.000002	1.481902	20.400000	8.300000	74.780000	80.700000	907.370558	DEFICIT
H	0.016274	0.001239	0.000007	0.624266	34.580000	20.920000	79.060000	60.280000	373.177745	DEFICIT
H	0.019919	0.001517	0.000008	0.764099	28.600000	20.140000	87.600000	81.620000	456.767831	DEFICIT

(source: CBS,MOAD,DHM,DFTQC,2017. See Table 3.1 for details)

f) Data Vectorization

Input data consists of string on the first column that is ecological Region. It consists of three classes H, M and T. H indicates Hill region, M indicates Mountain Region and T indicates Terai Region. Since there are string variables in the first and last column, these strings are encoded into numeric by using Label Encoder. It will encode different labels in that column with values between 0 to n-classes-1. In the above case, hill was replaced by 0, mountain by 1 and Terai by 2. Similarly, in case of last column, Deficit was replaced by 0, Normal was replaced by 1 and Surplus was replaced by 2. But these numbers do not mean that class 2 is greater than class 1.Thus, there was a need to create a dummy variable. When modeling multi-class classification problems using neural networks, it is good practice to reshape the output attribute from a vector that contains values for each class value to be a matrix with a boolean for each class value and whether or not a given instance has that class value or not. This is called one hot encoding or creating dummy variables from a categorical variable. For creating dummy variable "One hot encoding" was used. The input variable Region and output variable Production contains string values. For example, in this problem three class values are H, M and T. Turning this into a one-hot encoded binary matrix for each data instance would look as follows:

H - 0 0

M - 0 1

T - 1 0

g) Data Standardization

Each data was on a different scale, therefore, the data was standardized so that they all take small values on a similar scale. Mean and standard deviation of training data was calculated. The training data was standardized by subtracting the mean of each training data and dividing by the standard deviation. The same mean and standard deviation of training data was used to standardize the test data as well.

x = training data

y = testing data

$\bar{x}$ = mean of training data

σ = standard deviation of training data

$$\bar{x} = \frac{\sum x}{N} \quad \sigma = \sqrt{\frac{\sum(x-\bar{x}\ )^2}{N-1}} \qquad \text{… (equation 3.2.4)}$$

where, $\sum$x = sum of training data

N = number of training data

Standardized x and y data are calculated as:

$$x = \frac{x-\bar{x}}{\sigma} \qquad \text{… (equation 3.2.5)}$$

$$y = \frac{y-\bar{x}}{\sigma} \qquad \text{… (equation 3.2.6)}$$

3.2.3. Input Data

In total production data of total 1950 of 26 years data, there are 1266 deficit data, 137 normal data and 547 surplus data. This data shows that there is so much poor management of food production in Nepal such that Normal class is very low and food deficit dominates the food production of Nepal. Due to low number of Normal production, data for Normal class is low. This data was used as an input into Recurrent Neural Network.

3.3. Recurrent Neural Network (RNN)

RNN is a type of neural network that has an internal loop. Sequences are processed by iterating through the sequence elements and maintaining a state containing information relative to what it has seen so far. Example: Suppose while watching a movie, we keep watching the movie as at any point in time, we have the context because we have seen the movie until that point, then only we are able to relate everything correctly. It means, everything that is watched is remembered. Similarly, RNN remembers everything. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other. For eg: When it is needed to predict the next word in a given sentence, in that case, the relation among all the previous words helps in predicting the better output. The RNN remembers all these relations while training itself. In order to achieve it, the RNN creates the networks with loops in them, which allows it to persist the information. An unrolled version of RNN is shown in the Figure 3.2:

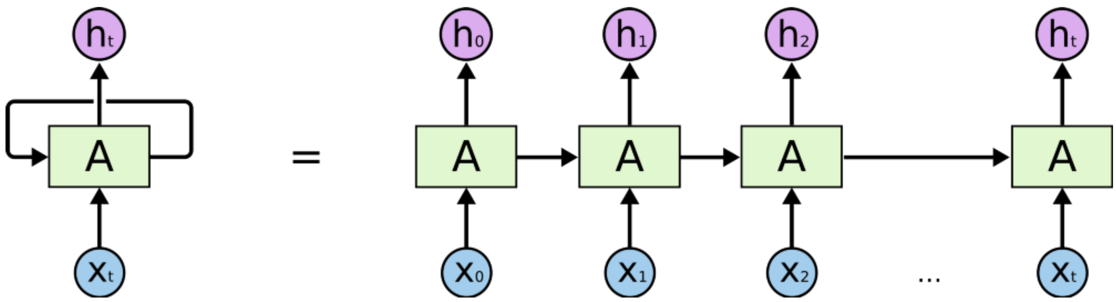


Figure 3.2: An Unrolled Simple RNN over time

$X_0, X_1 \ldots X_t$	= Sequence of input sample (each district (sample) with 10 variables)
$H_0, H_1 \ldots . H_t$	= Output of the sequence (Production Class)
A	= Holds information for the previous input samples (GRU)

Figure 3.2 shows that, at first, RNN takes X_0 from the sequence of input and then it outputs h_0 which together with X_1 is the input for the next step. So, h_0 and X_1 is the input for the next step. Similarly, h_1 from the next is the input with X_2 for the next step and so on. This way, it keeps remembering the context while training. RNN can be used wherever context from the previous input is needed.

Recurrent nets have predictive capacity. They grasp the structure of data dynamically over time, and they are used to predict the next element in a series. Those elements might be the next letters in a word, or the next words in a sentence (natural language generation); the next number in data from sensors, economic tables, stock price action, etc.

3.4. Gated Recurrent Unit (GRU)

GRUs are a gating mechanism in RNN, introduced by Chung et al. [16]. Its internal structure is simpler, and therefore is faster to train, since fewer computations are needed to make updates to its hidden state. To solve the vanishing gradient problem of a standard RNN, GRU uses update gate and reset gate [17]. GRU layers are somewhat streamlined and thus cheaper to run. GRUs have been shown to exhibit better performance on smaller datasets [18]. GRUs have fewer parameters and thus are faster to train and need less data to generalize [16]. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction [17]. The gates for a GRU cell are illustrated in the Figure 3.3

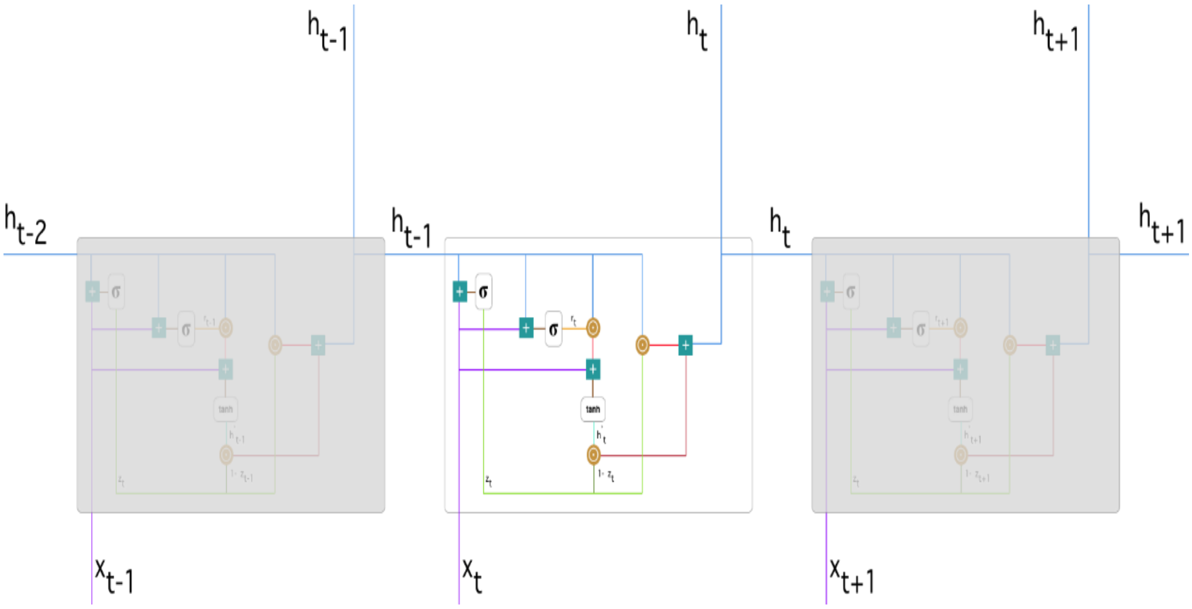


Figure 3.3: RNN with GRU

A single unit of RNN is shown in Figure 3.4.

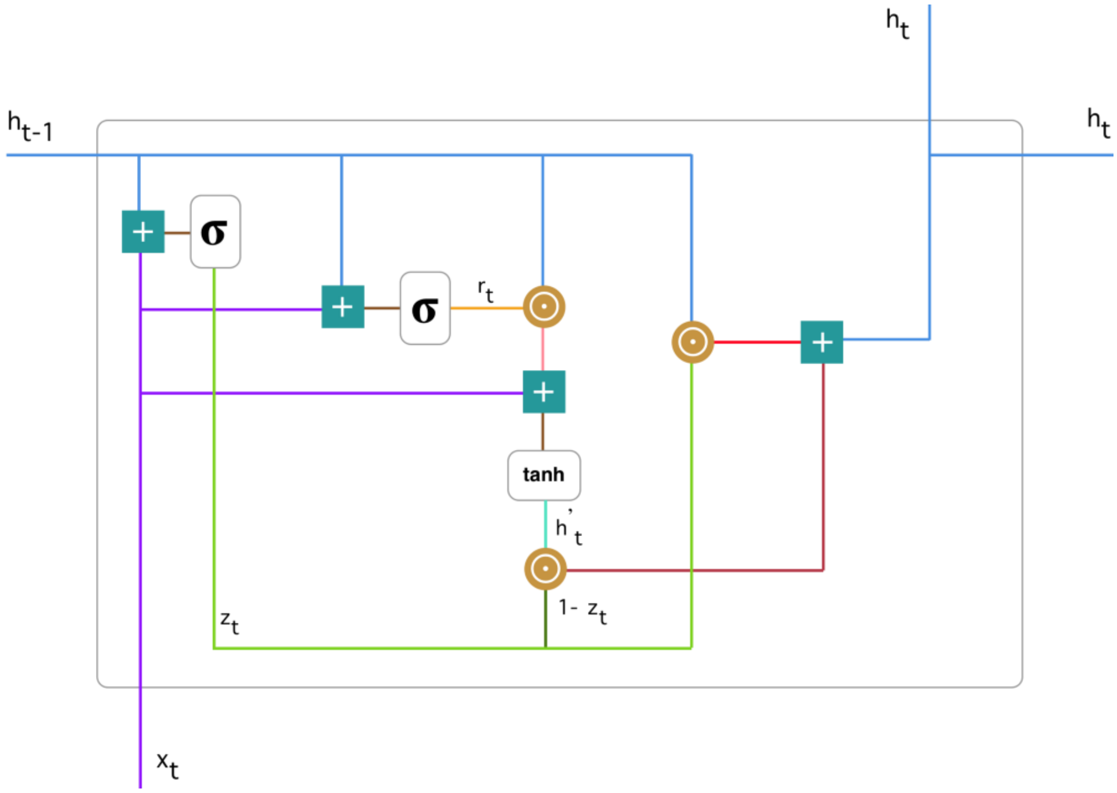


Figure 3.4: Single Unit of RNN with GRU

Introducing the notations:

"plus" operation "sigmoid" function "Hadamard product" operation "tanh" function

where,

x_t	= Input vector
h_t	= Hidden Layer Vector
z_t	= Update Gate
r_t	= Reset Gate
h_{t-1}	= Previous Hidden Layer Vector
h'_t	= Current Memory Content
W	= Weight of x_t
U	= Weight of h_t
σ	= Activation Function

There are two main gates update gate and reset gate in GRU which are described below:

a) Update gate

Update gate z_t is calculated for time step t using the formula:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} \qquad \text{...(equation 3.2.7)}$$

When x_t is plugged into the network unit, it is multiplied by its own weight $W^{(z)}$. The same goes for h_{t-1} which holds the information for the previous t-1 units and is multiplied by its own weight $U^{(z)}$. Both results are added together and a sigmoid activation function is applied to squash the result between 0 and 1 that's why called gates.

The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.

b) Reset gate

Essentially, this gate is used from the model to decide how much of the past information to forget. To calculate it, we use:

$$r_t = \sigma(W^{(r)} + U^{(r)}h_{t-1} \qquad \text{...(equation 3.2.8)}$$

This formula is the same as the one for the update gate. The difference comes in the weights and the gate's usage. As before, h_{t-1} and x_t are plugged in, multiplied with their corresponding weights, results are summed and sigmoid function is applied.

c) Current memory content

New memory content is introduced which will use the reset gate to store the relevant information from the past. It is calculated as follows:

$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1})$ ...(equation 3.2.9)

Input x_t is multiplied with it's weight W and h_{t-1} with it's weight U. Then, elementwise product is calculated between Reset r_t and weight U multiplied by h_{t-1}. That will determine what to remove from the previous time steps. Both the results are summed up and then non-linear activation function tanh is used.

d) Final memory at current time step

As a last step, the network needs to calculate h_t, vector which holds information for the current unit and passes it down to the network. In order to do that the update gate is needed. It determines what to collect from the current memory content, h_t and what from the previous steps h_{t-1}. That is done as follows:

$h = z_t \odot h_{t-1} + (1 - z) \odot h'_t$ …equation(3.2.10)

Following through, it can be seen how z_t is used to calculate 1-z_t which, combined with h'_t produces a result. z_t is also used with h_{t-1} in an element-wise multiplication. Finally, h_t is a result of the summation of the outputs.

It can be seen how GRUs are able to store and filter information using their update and reset gates. That eliminates the vanishing gradient problem since the model is not washing out the new input every single time but keeps the relevant information and passes it down to the next time steps of the network**.**

3.4.1. Initialization

To use Recurrent Neural Network (RNN) for time series modeling, it is essential to properly initialize the network, that is, to set the hidden neuron outputs properly at the initial time. RNN is initialized with zero state values or at steady state. In the context of dynamic system identification, such initializations imply the system to be modelled is in steady state, i.e., capturing transient behavior of the system is difficult if the network states are not properly initialized.

3.4.2. Activation function

A linear equation is easy to solve but they are limited in their complexity and have less power to learn complex functional mappings from data. To learn and represent almost anything and any arbitrary complex function which maps inputs to outputs and in order to get access to a much richer hypothesis space that would benefit from deep representations, a non-linearity is needed, or activation function. It makes the network more powerful and add ability to it to learn something complex and complicated form data and represent non-linear complex arbitrary functional mappings between inputs and outputs. Hence using a non-linear Activation non-linear mappings can be generated from inputs to outputs [19].

GRU use sigmoid function as activation function, and the cell recurrent connections use hyperbolic tangent function as activation function.

a) Sigmoid Activation function: It is an activation function of form

$$f(x) = \frac{1}{1+e^{-x}} \qquad \text{...equation(3.2.11)}$$

Its Range is between 0 and 1. It is a S-shaped curve.

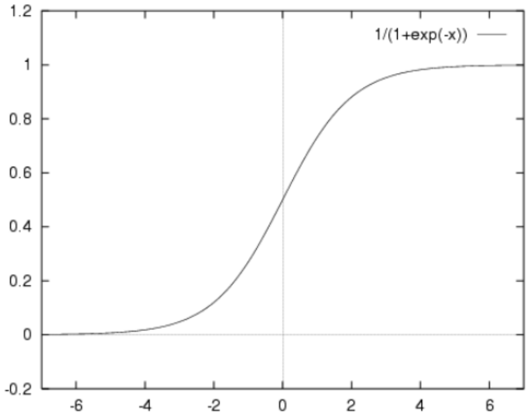


Figure 3.5: Sigmoid function

b) Hyperbolic Tangent activation function (Tanh) : It is an activation function of form

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \qquad \text{...equation(3.2.12)}$$

Its output is zero centered because its range in between -1 to 1

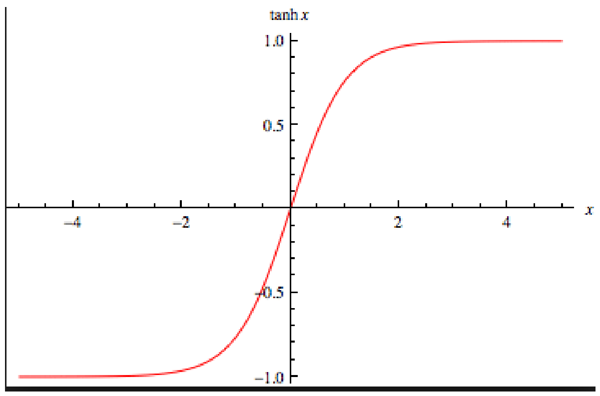


Figure 3.6: Hyperbolic Tangent Function

The cell recurrent connection need a function whose second derivative sustain for a long span to address the vanishing gradient problem. The gate recurrent connections could also use such a function, but since they control the error flow, in both positive and negative way, they use sigmoid as non-linearity [20].

c) Softmax activation function

The softmax activation function was used in the final layer of a neural network-based classifier. Such networks are commonly trained under a log loss (or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression. The softmax function is a generalization of the logistic function that "squashes" a K-dimensional vector z of arbitrary real values to a K-dimensional vector σ(z) of real values in the range [0,1] that add up to 1.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K \qquad \text{…equation(3.2.13)}$$

3.4.3. Loss function

To control the output of a neural network, output must be measured how far it is from what is expected. This is the job of the loss function of the network, also called the objective function. The loss function takes the predictions of the network and the true target and computes a distance score, capturing how well the network has done [18].

Classification is the problem of classifying instances into one of three or more classes. This is a multiclass classification problem meaning that there are three classes to be predicted i.e Surplus, Deficit and Normal. The best loss function to use in this case is

Categorical Cross Entropy. It measures the distance between two probability distributions: here, between the probability distribution output by the network and the true distribution of the labels. By minimizing the distance between these two distributions, the network is trained to output something as close as possible to the true labels [18].

$$H(p, q) = -\sum_x p(x)\log(q(x)) \qquad \text{...equation(3.2.14)}$$

p(x) = output of the network

q(x) = true distribution of the labels

3.4.4. Optimizer function

Loss score is used as a feedback signal to adjust the value of the weights a little, in a direction that will lower the loss score. This adjustment is the job of the optimizer. RMS optimizer is usually a good choice for recurrent neural networks. RmsProp is an optimizer that utilizes the magnitude of recent gradients to normalize the gradients. A moving average is always kept over the root mean squared gradients, by which the current gradient is divided. It has an effect of balancing the step size i.e. decrease the step for large gradient to avoid exploding, and increase the step for small gradient to avoid vanishing.

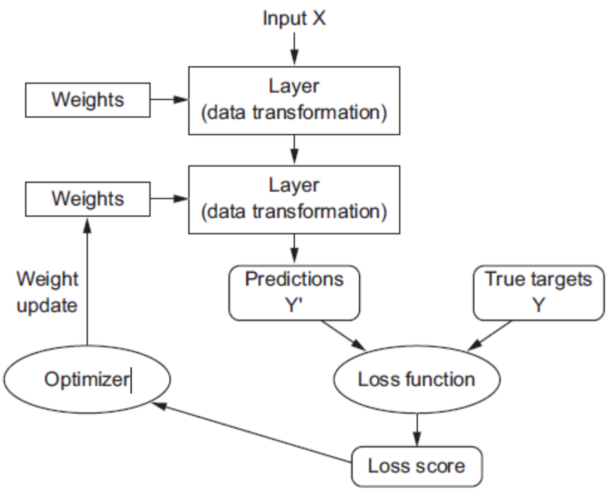


Figure 3.7: Loss and Optimizer function

3.4.5. Split Data Selection

Different train/test split ratio were tested by seeing the loss epoch curve to decide the best split for data. Split of 50:50 i.e. 50% of training and 50% of testing data produced loss of 0.6225, 60:40 produced loss of 0.5893, 70:30 produced loss of 0.546 and 80:20 produced loss of 0.5969. From the results, it can be concluded that under fitting occurred for splits 50:50 and 60:40 because loss was continuously decreasing until split 70:30. Then over fitting occurred after 70:30 because at 80:20 loss started increasing. So, split data 70:30 was selected because it produced minimum loss amongst all splits. Tabulated loss data of train/test split is shown in Annex B. The graph of loss comparison of different splits is shown in Figure 3.8:

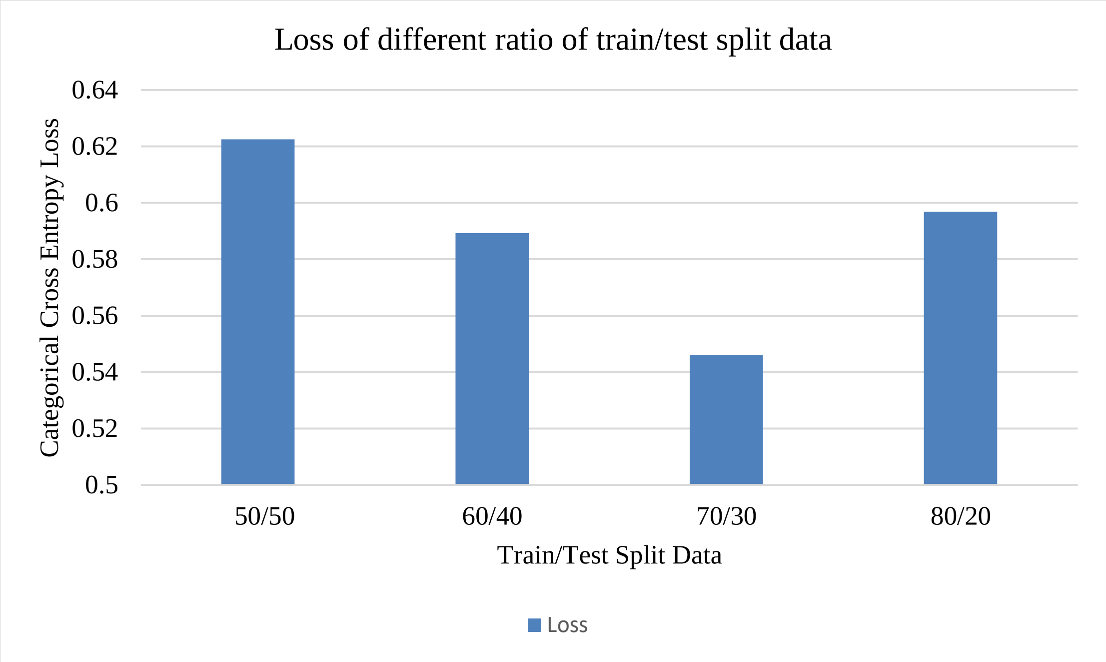


Figure 3.8: Comparison of loss of different train/test split data

3.4.6. Hyper-parameter Selection from validation data

Since there is no pre-defined rule for deciding hyper-parameters such as the number of neurons, size of the architecture or learning rate, optimum model was selected by seeing the loss-epoch curve and deciding what number of neurons and size of architecture performs best for data used in this study. Different hyper-parameters were tested in this study to decide which hyper-parameter produces best result. The model was trained on 70% of the data. 70% training data was further split in 70% data for training and 30% data for validation to select optimum hyper-parameter for the GRU model.

Some of the tabulation of the loss-epoch curve of some of the hyper-parameters used is shown in Table 3.5. More learning rates, number of neurons and size of neurons that were tested to decide the optimum model for prediction are shown in Annex B.

Table 3.5: Loss and Accuracy of GRU model using different hyper-parameters

LR	Neurons	Validation Loss and Accuracy
0.01	3	Epoch 33/1000 - 0s - loss: 0.3407 - acc: 0.9108 - val_loss: 0.5978 - val_acc: 0.8187
0.01	10	Epoch 69/1000 - 0s - loss: 0.3014 - acc: 0.9118 - val_loss: 0.5455 - val_acc: 0.8213
0.01	23	Epoch 96/1000 - 0s - loss: 0.2440 - acc: 0.9149 - val_loss: 0.5165 - val_acc: 0.8160
0.01	24	Epoch 83/1000 - 0s - loss: 0.1986 - acc: 0.9231 - val_loss: 0.5321 - val_acc: 0.8267
0.001	23	Epoch 386/1000 - 0s - loss: 0.2612 - acc: 0.9149 - val_loss: 0.5346 - val_acc: 0.8187
0.1	23	Epoch 722/1000 - 0s - loss: 0.3695 - acc: 0.9087 - val_loss: 0.5538 - val_acc: 0.8107
0.009	23	Epoch 96/1000 - 0s - loss: 0.2649 - acc: 0.9108 - val_loss: 0.5391 - val_acc: 0.8240
0.02	23	Epoch 27/1000 - 0s - loss: 0.2744 - acc: 0.9138 - val_loss: 0.5572 - val_acc: 0.8213

Table 3.5 shows that at epoch 96, using 23 neurons and 1 hidden layer, the model produced minimum validation loss of 0.5165 and accuracy of 81.60% using learning rate of 0.01 at epoch 96, so it was chosen. Increasing number of hidden layers or number of neurons in the hidden layer did not improve the result.

Epoch 96/1000

- 0s - loss: 0.2440 - acc: 0.9149 - val_loss: 0.5165 - val_acc: 0.8160

The optimum loss epoch curve is shown in the Figure 3.9. Loss epoch curve of some of the hyper-parameters is shown in Annex C.

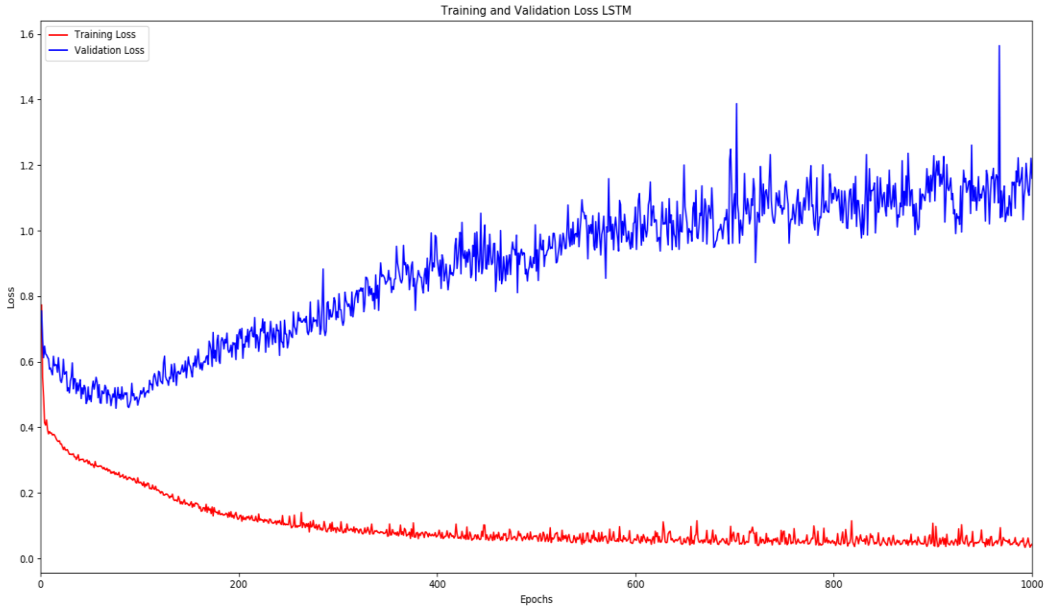


Figure 3.9: Loss epoch curve of optimum hyper-parameters of GRU model

3.4.7. Architecture of Optimized GRU model

Input Variables consists of 10 variables namely ecological Region, Nitrogen, Phosphorous, Potassium, Compost, Maximum Temperature, Minimum Temperature, Relative Humidity (RH) morning, Relative Humidity (RH) evening and Rainfall. The output consists prediction of one of the three classes namely Deficit, Surplus or Normal. The model outputs one of the three classes based on the distance between probability distribution output by the network and the true distribution of the labels. The GRU architecture developed using hyper-parameters from above loss epoch curve is shown in Figure 3.10. The GRU model consists of one hidden layer with 23 neurons using learning rate of 0.01. In Figure 3.10, each unit in the hidden layer is a GRU unit.

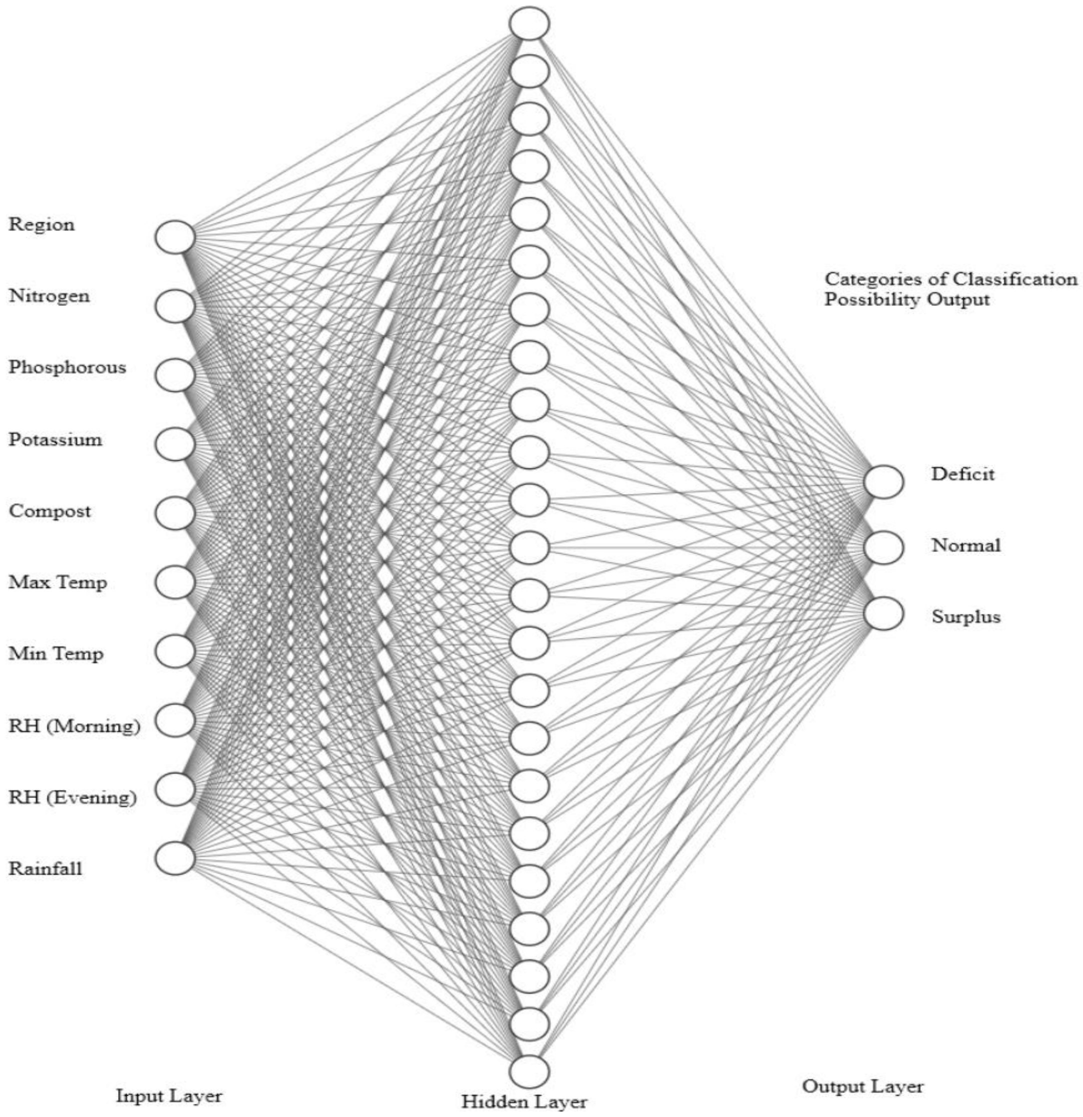


Figure 3.10: Optimized GRU model for prediction

3.4.8. Accuracy Metrics

The study used following accuracy metrics for further evaluating accuracy of the optimized GRU model.

a) Confusion Matrix

A clean and unambiguous way to present the prediction results of a classifier is to use a confusion matrix (also called a contingency table). For a binary classification problem the table has 2 rows and 2 columns. Across the top is the observed class labels and down the side are the predicted class labels. Each cell contains the number of predictions made by the classifier that fall into that cell. The confusion matrix is shown in Table 3.6.

Table 3.6: Confusion Matrix

Actual/Predicted	Positive	Negative
Positive	True Positive(TP)	False Negative(FN)
Negative	False Positive(FP)	True Negative(TN)

A perfect classifier would correctly predict no recurrence and recurrence which would be entered into the bottom right cell no recurrence/no recurrence called True Negatives (TN) and top left cell recurrence/recurrence called True Positives (TP). Incorrect predictions are clearly broken down into the two other cells. False Negatives (FN) which are recurrence that the classifier has marked as no recurrence. False Positives (FP) are no recurrence that the classifier has marked as recurrence.

For multiclass classification,

- TP is the value in the main diagonal.
- FN for each class is the sum of all values in the corresponding row excluding (TP).
- FP for each class is the sum of all values in the corresponding column excluding the main diagonal element (TP).
- TN for each class is the sum of all the values of the confusion matrix excluding that class's row and column.

b) Precision

Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV).

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

c) Recall

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. It is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

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

d) F1 Score

The F1 Score is also called the F Score or the F Measure. The F1 score conveys the balance between the precision and the recall.

$$\text{F1} = 2 * \frac{\text{Precision. Recall}}{\text{Precision} + \text{Recall}}$$

3.4.9. Baseline ANN Model Comparison with GRU model

Baseline ANN model was trained on Siraha district using data of thirteen years starting from 2001 to 2013 using six input variables. Different ecology was not considered since prediction was done only for Siraha district.

Proposed study used GRU model and GRU model was trained on 75 districts of Nepal using data of 26 years starting from 1991-2016 using ten input variables. Since all the districts were used for training and testing, different ecology was considered.

Tabulated comparison of Baseline ANN model used in Nepal with proposed GRU model is shown in Table 3.7:

Table 3.7: Features of Baseline ANN model [1]

Crop	Rice
Type of NN	ANN Backpropagation
District data used	Siraha
Duration of data	13 years (2001-2013)
Input Variables Used	6
Different Ecology Considered	No

Table 3.8: Features of proposed study using GRU model

Crop	Rice
Type of NN	RNN GRU model
District data used	75
Duration of data	26 years (1991-2016)
Input Variables Used	10
Different Ecology Considered	Yes

To compare GRU RNN with baseline ANN, ANN architecture was recreated by using the same baseline architecture as used by Ranjeet & Armstrong [6] as shown in Figure 3.11.

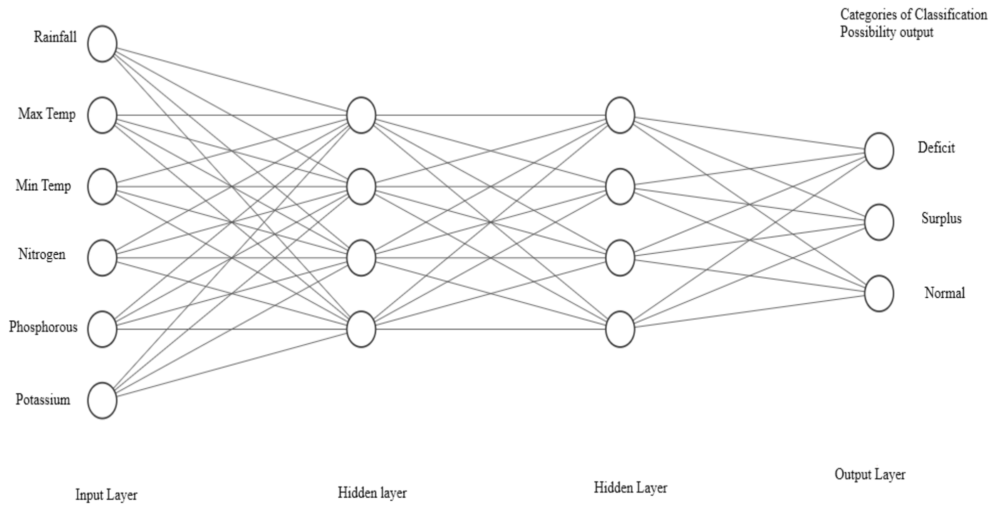


Figure 3.11: GRU RNN model with 6 input variables

Figure 3.11 shows the baseline ANN model with 6 input variables. The 6 input variables are Rainfall, Maximum Temperature, Minimum Temperature, Nitrogen, Potassium and Phosphorous. These variables were used from thirteen years starting from 2001 to 2013. 2 hidden layers with 4 neurons in each layer, using tanh as activation function, which forecasts which class Production belongs to.

Then the same architecture and same variables were used for testing with GRU RNN. The both networks were trained with 8 years of data and tested on 5 years of data.

Then using the same number of hidden layers and neurons, the input variables were increased to 10. The 10 variables used were ecological Region, Nitrogen, Phosphorous, Potassium, Compost, Maximum Temperature, Minimum Temperature, RH (morning), RH (Evening), and Rainfall. The architecture can be seen in Figure 3.12:

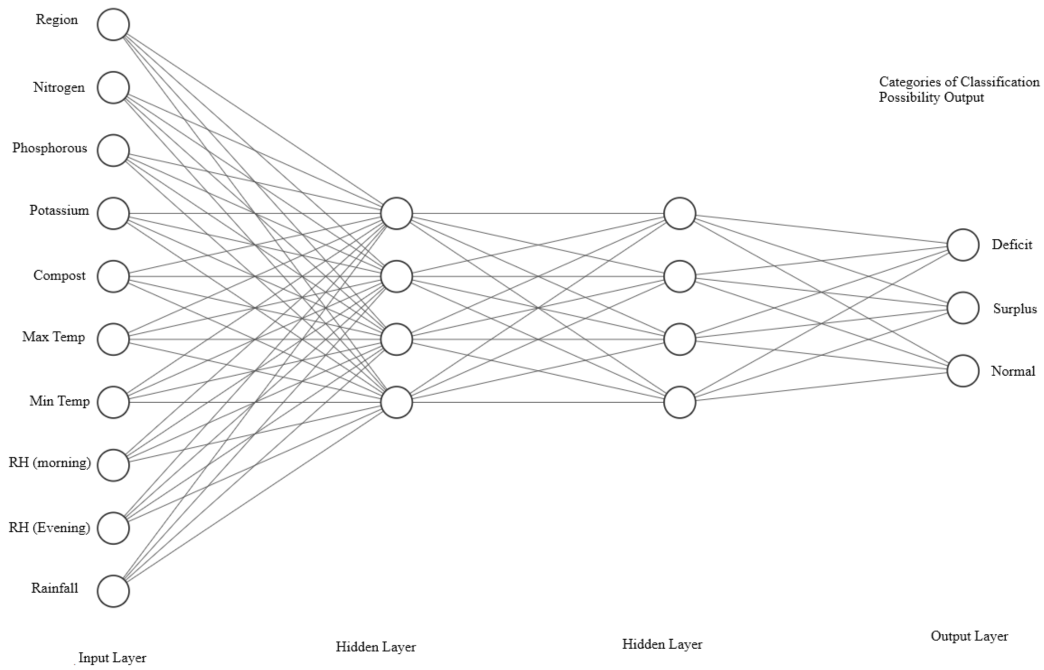


Figure 3.12: RNN model with 10 variables

3.5. Software and Tools used

The study used following programming language and hardware to test the optimized GRU model. The tools used in the study are shown in Table 3.9

Table 3.9: Tools used in the study

Item	Tool
Operating System	Windows 10 64-bit
Processor	Intel® Core™ i7-7700HQ CPU @ 2.80 GHz
Random Access Memory	8 GB
Graphics Processing Unit	GeForce GTX 1050 ti 4 GB
Programming Language	Python

CHAPTER IV: RESULTS AND DISCUSSION

4.1. Results

4.1.1. Results of GRU model for testing on 30% data

The optimized GRU model with learning rate of 0.01, 1 hidden layer and 23 neurons on the hidden layer was trained on 70% of data and used to test on remaining 30% of the data set which gave testing classification accuracy of 81% with loss of 0.5460.

Confusion matrix was used to further evaluate the performance of the GRU model. Confusion Matrix of the optimum GRU model is shown in Table 4.1.
From Table 3.5, hyper-parameters that gave least loss were selected which is given below:

Learning rate=0.01

Number of Hidden Layers = 1

Number of Neurons = 23

Table 4.1: Confusion Matrix of Optimum GRU Model

Actual	Predicted			
	Surplus	Deficit	Normal	Total
Surplus	125	40	5	170
Deficit	23	355	9	387
Normal	20	17	6	43

Table 4.1 shows that Deficit class dominates majority of the data which is 387 out of 600 data followed by Surplus class which is 170. Normal class contains only 43 of the total data out of 600 data.

Then, TP, TN, FP and FN were calculated as shown in the Table 4.2.

Table 4.2: TP, TN, FP and FN of optimum GRU model

Classes	TP	TN	FP	FN
Surplus	125	387	43	45
Deficit	355	156	57	32
Normal	6	543	14	37

Using the values of TP, TN, FP and FN Precision, Recall and F1-score were calculated.

Table 4.3: Precision, Recall and F1-Score of optimum GRU model

Class	Precision	Recall	F1-Score
Surplus	0.744047619	0.735294	0.73964497
Deficit	0.861650485	0.917313	0.888610763
Normal	0.3	0.139535	0.19047619

Since normal class contains least amount of data out of total data, precision, recall and f1-score of the model for Normal class is low.

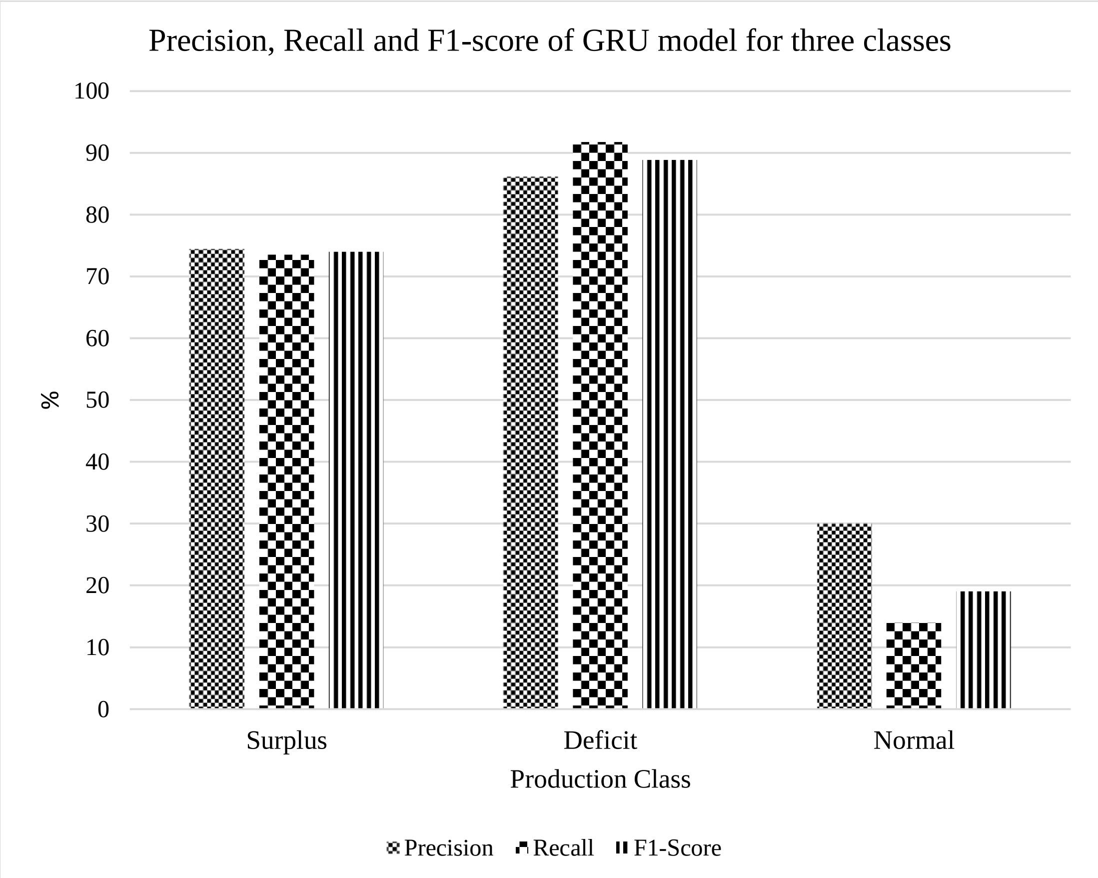


Figure 4.1: Precision, Recall and F1-score of the optimum GRU model

Further, Receiver Operating Characteristics (ROC) curve was created for different production classes by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) which is shown in Figure 4.2, 4.3 and 4.4.

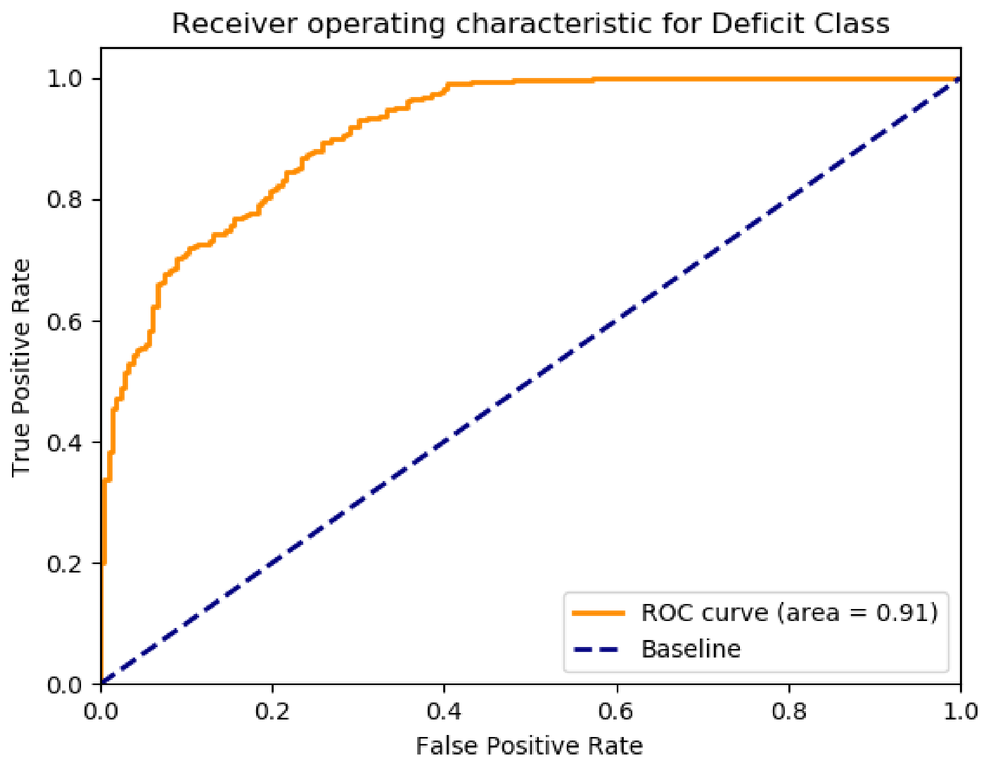


Figure 4.2: ROC curve for Deficit Class

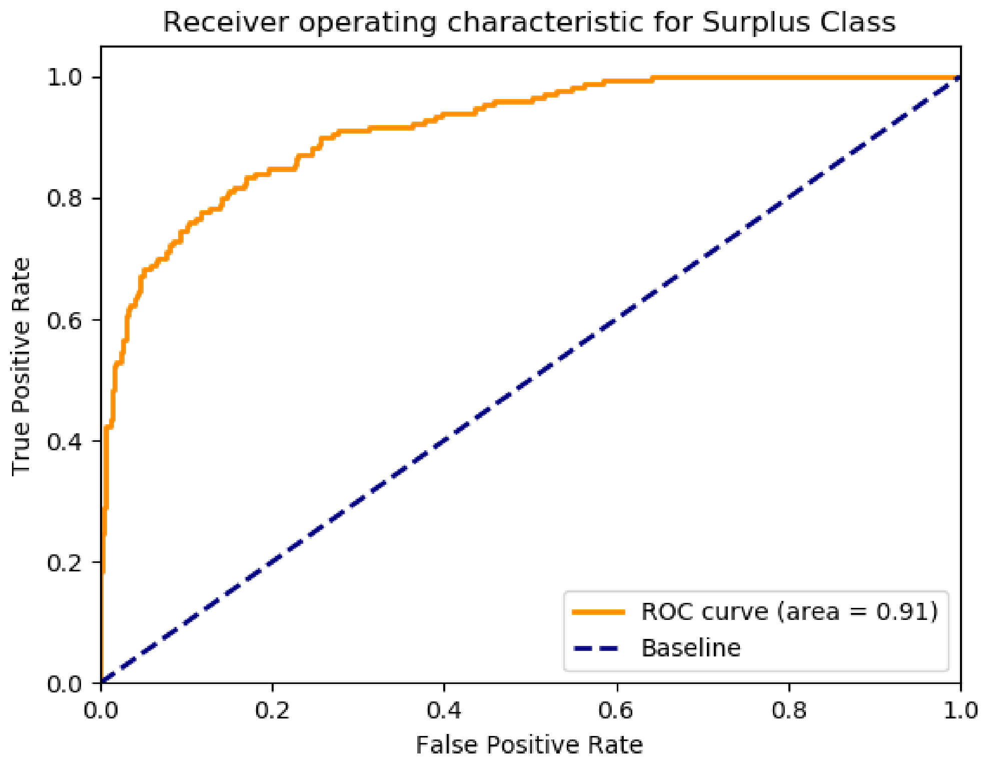


Figure 4.3: ROC curve for Surplus Class

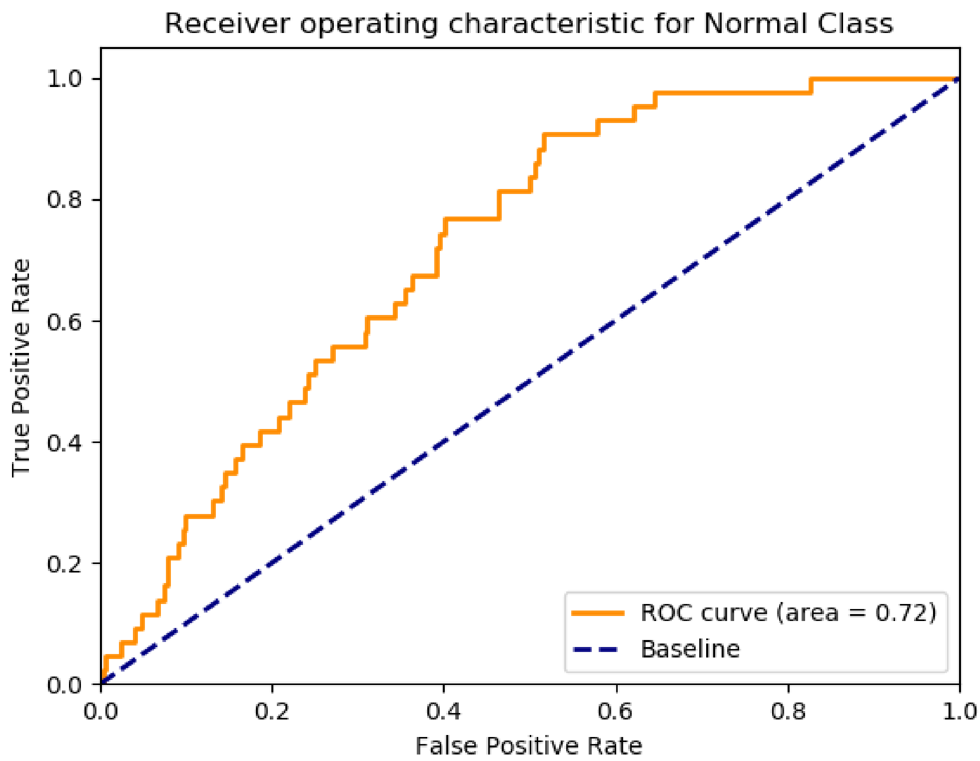
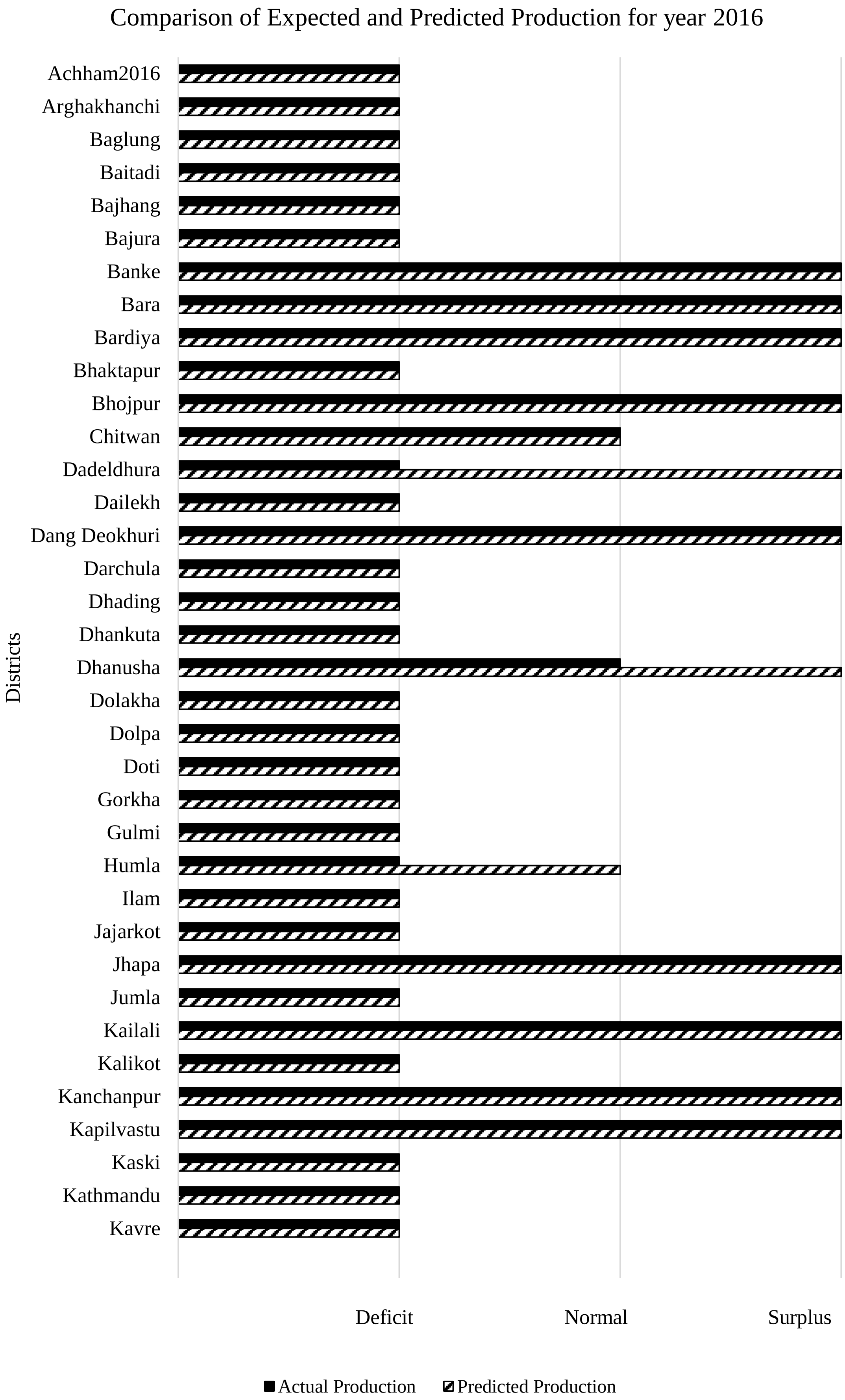


Figure 4.4: ROC curve for Normal Class

Figure 4.2 and 4.3 shows that Deficit and Surplus class cover total area of 0.91 but since Normal data was very low i.e 43 out of total 600, in Figure 4.4 Normal class nearly touches the baseline covering only area of 0.72.

4.1.2. Results of GRU model for year 2016

The GRU model was tested on year 2016. The results of GRU model for each district for year 2016 is presented in the Figure 4.5.



Comparison of Expected and Predicted Production for year 2016
Districts
Achham2016
Arghakhanchi
Baglung
Baitadi
Bajhang
Bajura
Banke
Bara
Bardiya
Bhaktapur
Bhojpur
Chitwan
Dadeldhura
Dailekh
Dang Deokhuri
Darchula
Dhading
Dhankuta
Dhanusha
Dolakha
Dolpa
Doti
Gorkha
Gulmi
Humla
Ilam
Jajarkot
Jhapa
Jumla
Kailali
Kalikot
Kanchanpur
Kapilvastu
Kaski
Kathmandu
Kavre
Deficit
Normal
Surplus
Actual Production
Predicted Production

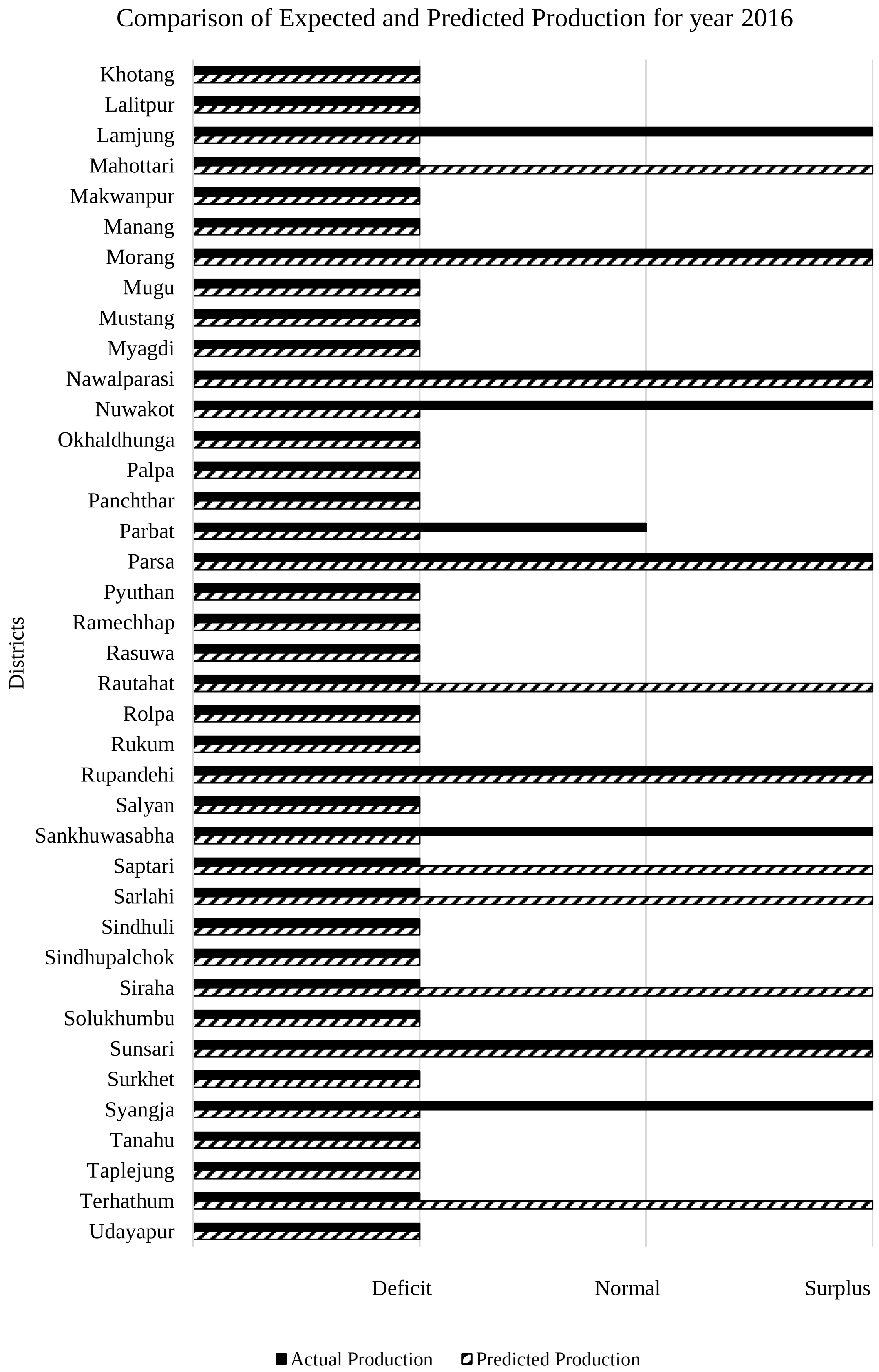


Figure 4.5: Comparison of Expected and Predicted Production for Year 2016

Figure 4.5 shows the comparison of expected and predicted production for the year 2016. The figure shows that in year 2016, there are total number of 18 Surplus data, 54 deficit data and 3 Normal data. Amongst them, the model correctly classified 14 surplus data, 46 deficit data and 1 Normal data. Remaining 14 data were misclassified. The accuracy of model for year 2016 was 81.33%. Precision and recall were further calculated to evaluate the accuracy of the model. Among 14 misclassified data, 11 data were misclassified as 2 step variation i.e. Surplus prediction for Deficit data or Deficit Prediction for Surplus data and 3 data were misclassified as one step variation i.e. Deficit prediction for Normal data or Normal prediction for Deficit data or Surplus prediction for Normal data. Since the number of Normal class data is very low, only one Normal class was predicted. The tabulation of the above graph is shown in Annex D.

Confusion matrix, TP, TN, FP and FN, F1-score, Recall and Precision is shown in the tables 4.4, 4.5 and 4.6 respectively.

Table 4.4: Confusion Matrix for year 2016

Actual	Predicted			Total
	Surplus	Deficit	Normal	
Surplus	14	4	0	18
Deficit	7	46	1	54
Normal	1	1	1	3

Table 4.5: TP, TN, FP and FN for year 2016

Class	TP	TN	FP	FN
Surplus	14	49	8	4
Deficit	46	16	5	8
Normal	1	71	1	2

Table 4.6: Precision, Recall and F1-Score for year 2016

Class	Precison	Recall	F1-Score
Surplus	0.636363636	0.777778	0.7
Deficit	0.901960784	0.851852	0.876190476
Normal	0.5	0.333333	0.4

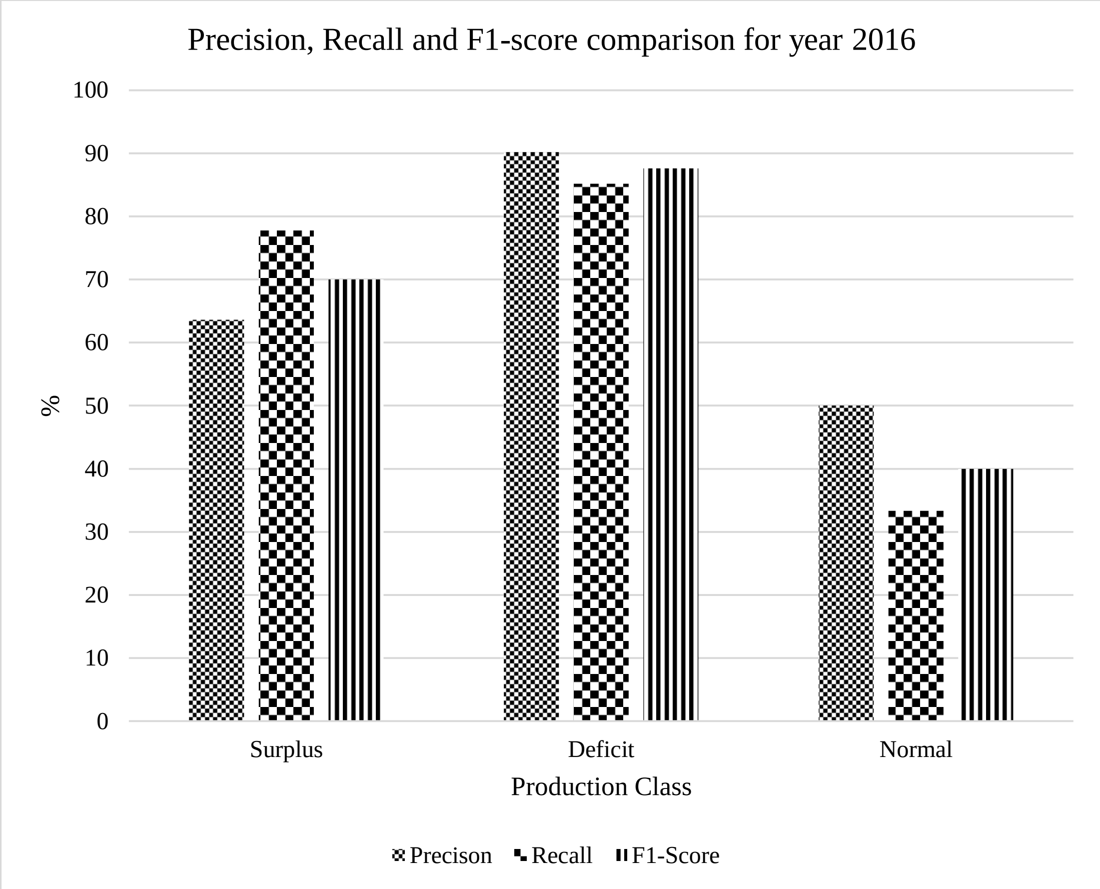
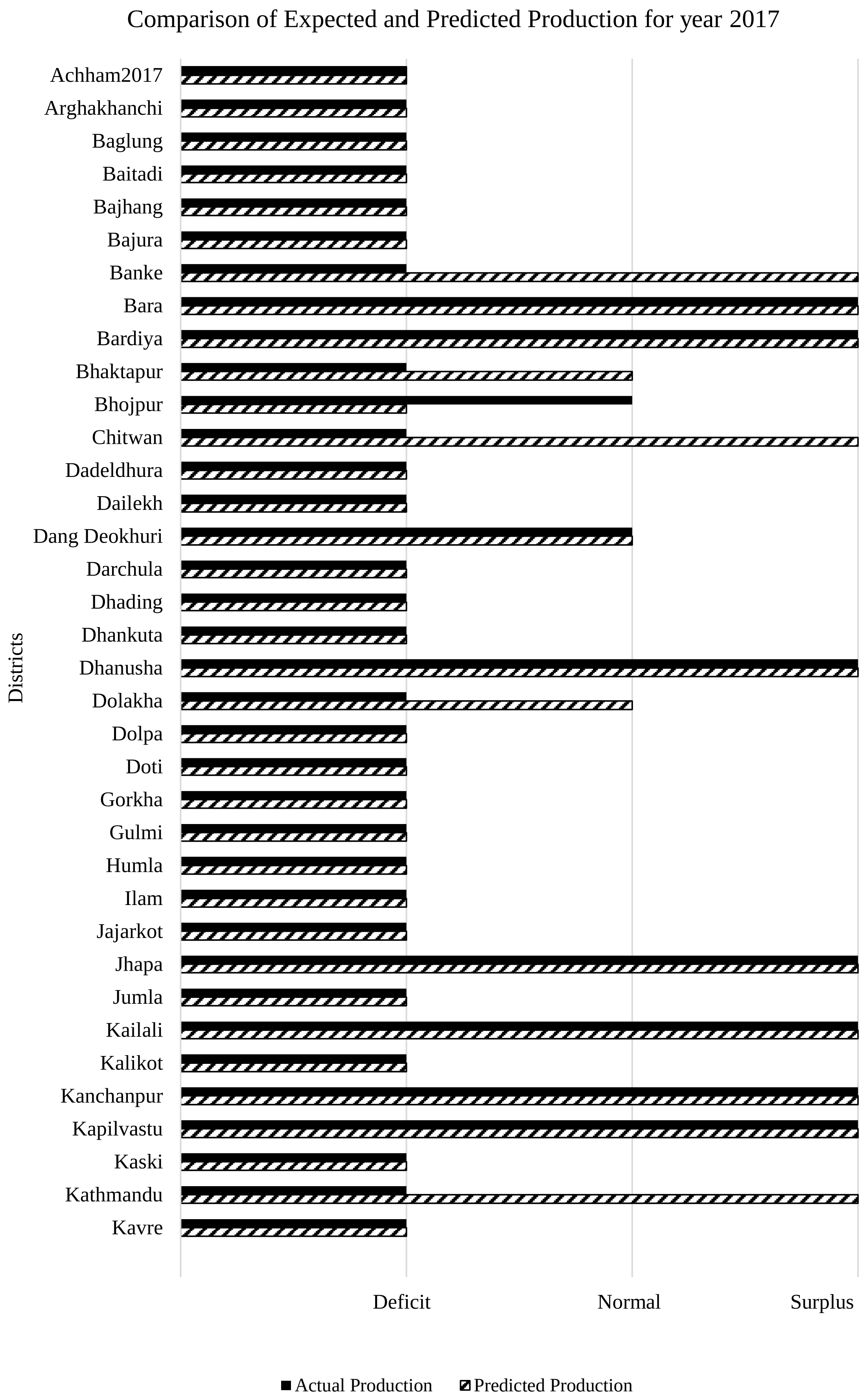


Figure 4.6: Precision, Recall and F1-Score of three classes for year 2016

Figure 4.6 shows the comparison Precision, Recall and F1-score of three classes for year 2016

4.1.3. Results of GRU model for year 2017

The GRU model was tested on year 2017. The results of GRU model for each district for year 2017 is presented in the Figure 4.7.



Comparison of Expected and Predicted Production for year 2017
Districts
Achham2017
Arghakhanchi
Baglung
Baitadi
Bajhang
Bajura
Banke
Bara
Bardiya
Bhaktapur
Bhojpur
Chitwan
Dadeldhura
Dailekh
Dang Deokhuri
Darchula
Dhading
Dhankuta
Dhanusha
Dolakha
Dolpa
Doti
Gorkha
Gulmi
Humla
Ilam
Jajarkot
Jhapa
Jumla
Kailali
Kalikot
Kanchanpur
Kapilvastu
Kaski
Kathmandu
Kavre
Deficit
Normal
Surplus
Actual Production
Predicted Production

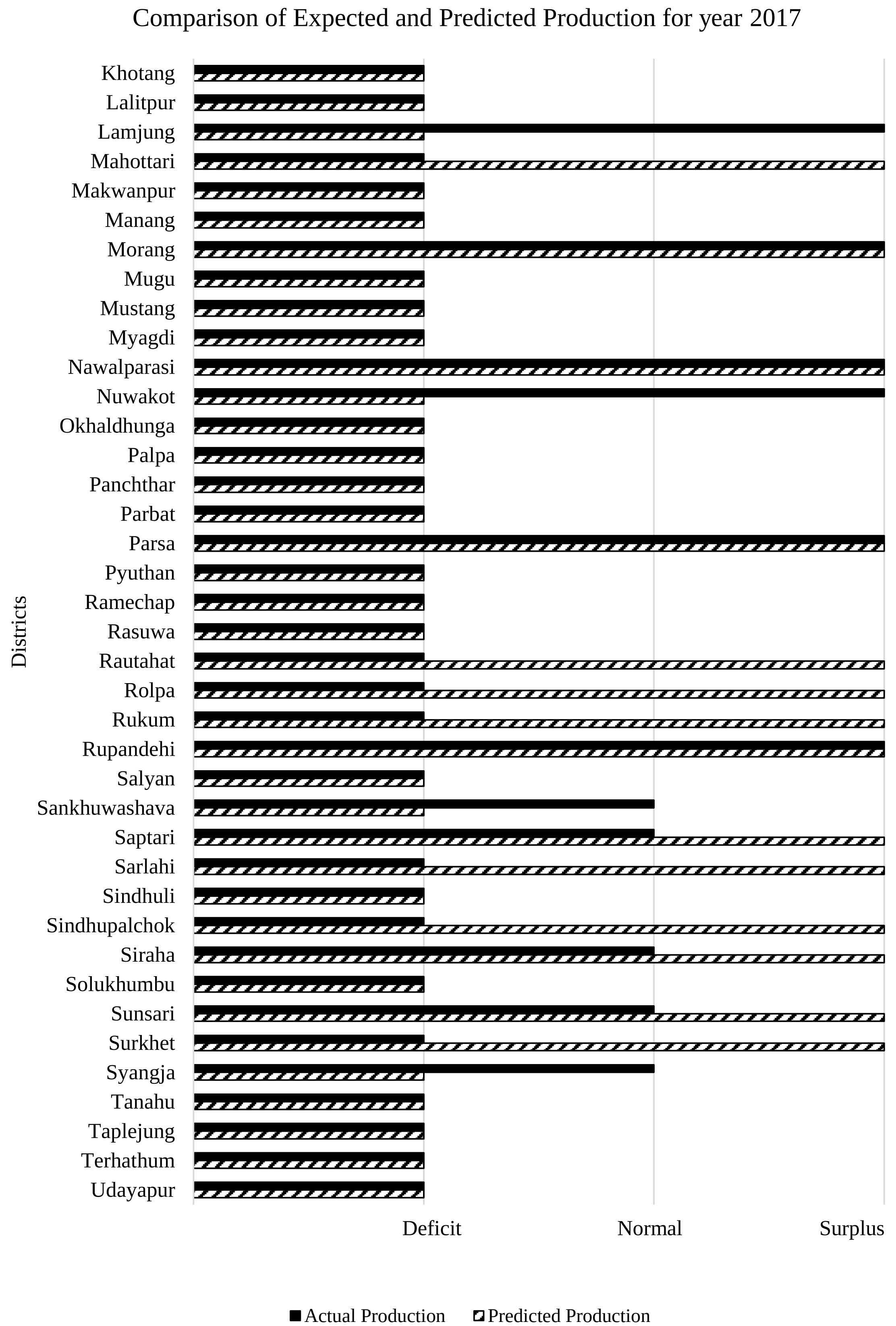


Figure 4.7: Comparison of Expected and Predicted Production for Year 2017

Figure 4.7 shows the comparison of expected and predicted production for the year 2017. The figure shows that in year 2017, there are total number of 13 Surplus data, 55 deficit data and 7 Normal data. Amongst them, the model correctly classified 11 surplus data, 43 deficit data and 1 Normal data. Remaining 20 data were misclassified. The accuracy of model for year 2017 is 73.33%. Precision and recall were further calculated to evaluate the accuracy of the model. Among 20 misclassified data, 12 data were misclassified as 2 step variation i.e. Surplus prediction for Deficit data or Deficit Prediction for Surplus data and 8 data were misclassified as one step variation i.e. Deficit prediction for Normal data or Normal prediction for Deficit data or Surplus prediction for Normal data. Since the number of Normal class data is very low, only one Normal class was predicted. The tabulation of the above graph is shown in Annex D.

Confusion matrix, TP, TN, FP and FN, F1-score, Recall and Precision is shown in the tables 4.7, 4.8 and 4.9 respectively.

Table 4.7: Confusion Matrix for year 2017

Actual	Predicted			Total
	Surplus	Deficit	Normal	
Surplus	11	2	0	13
Deficit	10	43	2	55
Normal	3	3	1	7

Table 4.8: TP, TN, FP and FN for year 2017

Class	TP	TN	FP	FN
Surplus	11	49	13	2
Deficit	43	15	5	12
Normal	1	66	2	6

Table 4.9: Precision, Recall and F1-Score for year 2017

Class	Precision	Recall	F1-Score
Surplus	0.458333333	0.846154	0.594594595
Deficit	0.895833333	0.781818	0.834951456
Normal	0.333333333	0.142857	0.2

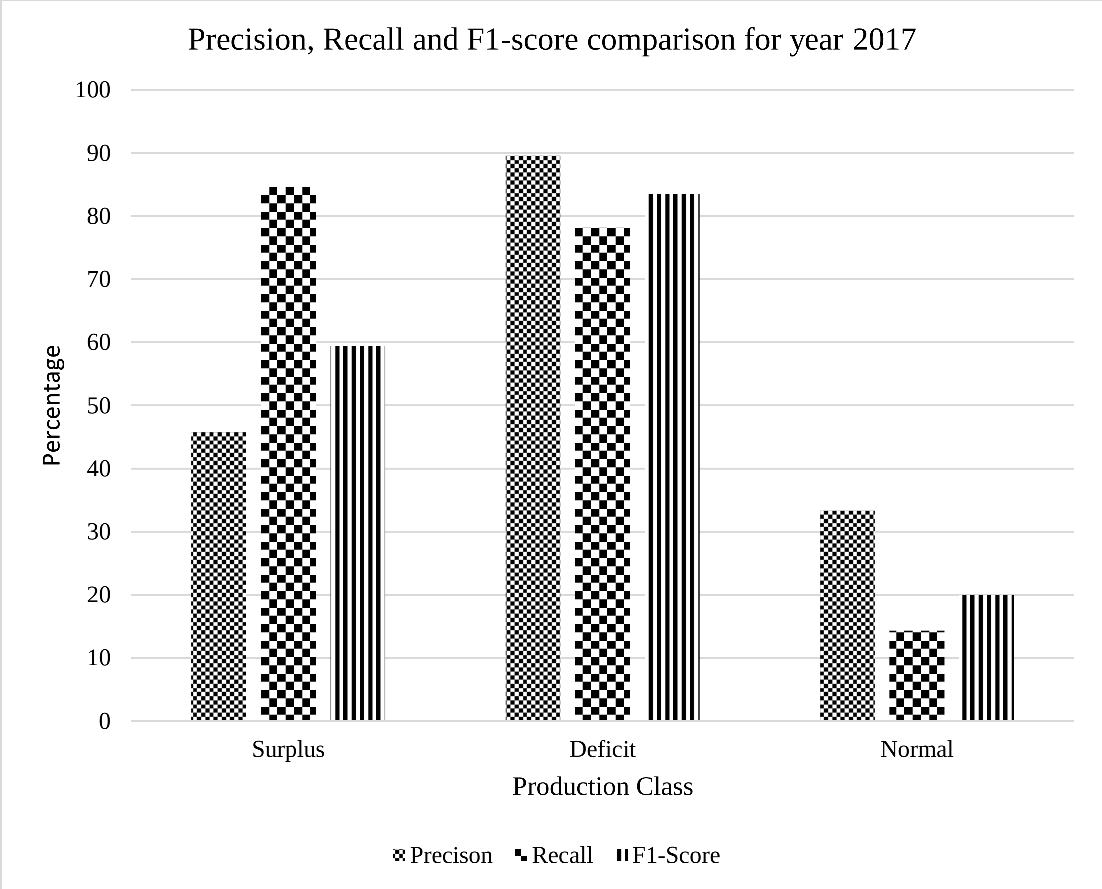


Figure 4.8: Precision, Recall and F1-Score of three classes for year 2017

Figure 4.8 shows the comparison Precision, Recall and F1-score of three classes for year 2017

4.1.4. Comparison of GRU model with baseline ANN model

Now to further validate that GRU model performs better than baseline ANN model, 6 input variables were used to compare GRU model with baseline ANN model. The 6 input variables were Rainfall, Maximum Temperature, Minimum temperature, Potassium, Phosphorous and Nitrogen. The models were trained on 8 years of data and tested on 5 years of data from Siraha District. Baseline ANN model achieved an accuracy of 80% with loss of 0. 4193. When RNN based GRU model was used, it could correctly classify all data with loss of 0.2420. Then using the same architecture but using all the 10 input variables, GRU model was tested again which could also correctly classify all of the data but loss drastically reduced to 0.0118. Graphical representation of accuracy and loss is shown in Figure 4.9 and Figure 4.10 respectively.

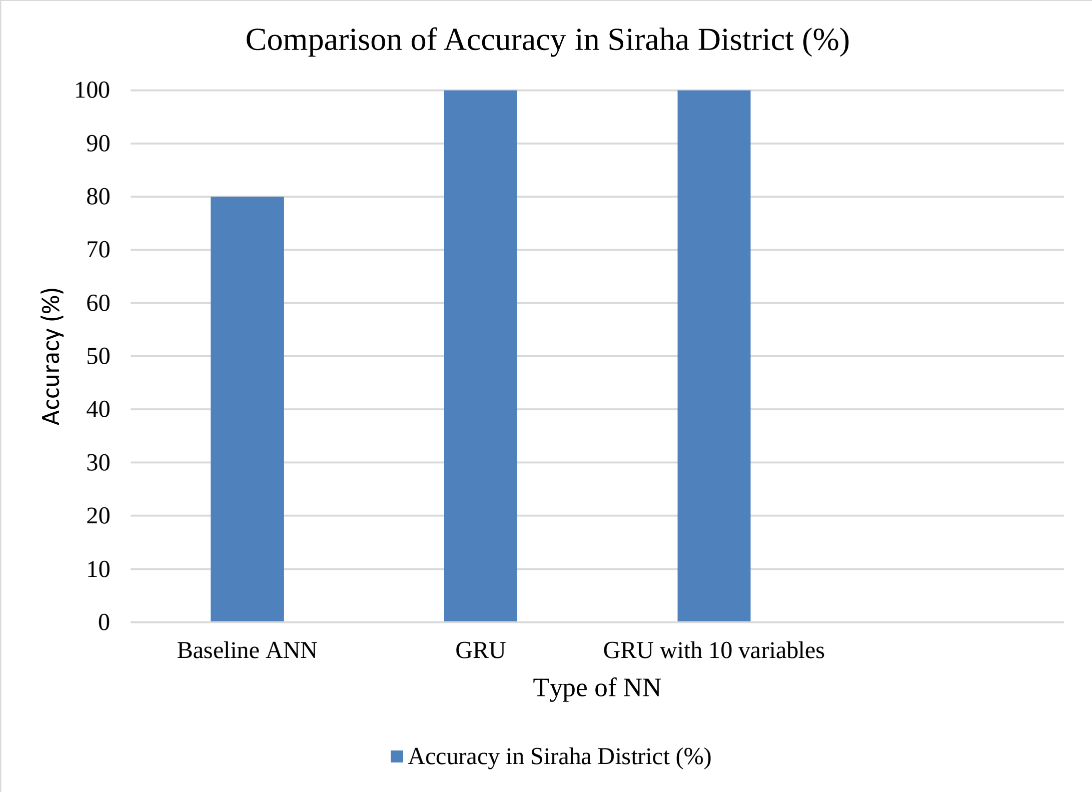


Figure 4.9: Comparison of accuracy of baseline ANN with RNN model

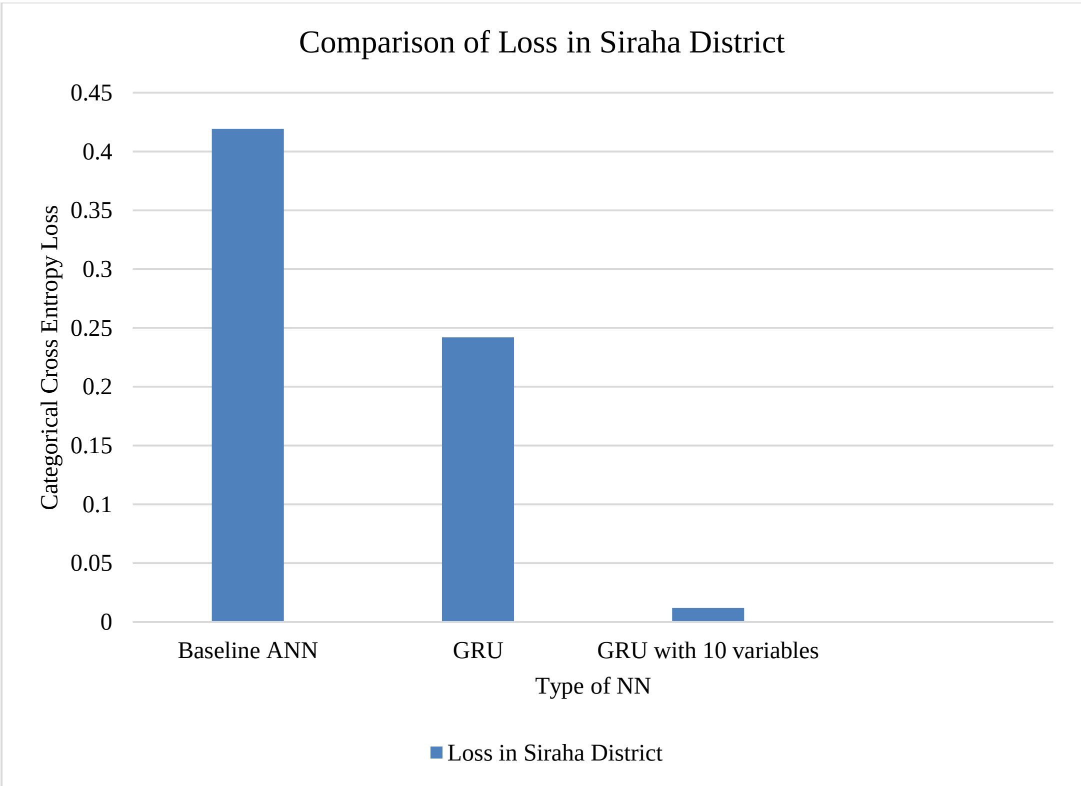


Figure 4.10: Comparison of loss of baseline ANN with GRU model

Confusion matrix, F1-score, Recall and Precision of Baseline ANN model for Siraha district is shown in the tables 4.10 and 4.11 respectively.

Table 4.10: Confusion Matrix of Baseline ANN model

Actual	Predicted			Total
	Surplus	Deficit	Normal	
Surplus	2	0	0	2
Deficit	0	1	0	1
Normal	1	0	1	2

Table 4.11: Precision, Recall and F1-Score of Baseline ANN model

Class	Precision	Recall	F1-Score
Surplus	1	1	1
Deficit	0.5	1	0.666666667
Normal	1	0.5	0.666666667

Confusion matrix, F1-score, Recall and Precision of GRU model for Siraha district is shown in the tables 4.12 and 4.13 respectively.

Table 4.12: Confusion Matrix of GRU model

Actual	Predicted			Total
	Surplus	Deficit	Normal	
Surplus	2	0	0	2
Deficit	0	1	0	1
Normal	0	0	2	2

Table 4.13: Precision, Recall and F1-Score of GRU model

Class	Precision	Recall	F1-Score
Surplus	1	1	1
Deficit	1	1	1
Normal	1	1	1

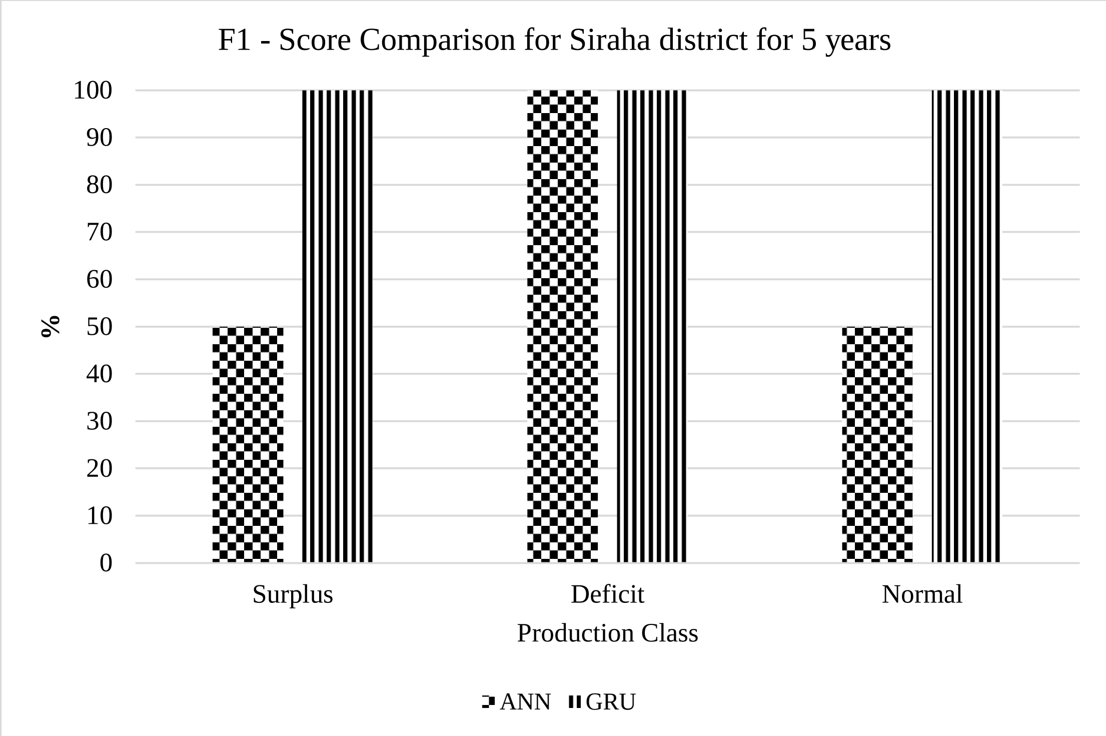


Figure 4.11: F1-Score of Baseline ANN and GRU model

Figure 4.11 shows the comparison of F1-Score for ANN and GRU model for different Production classes.

4.1.5. Comparison of Execution time of baseline ANN and GRU model

Baseline ANN model and GRU model were executed 10 times and the average execution of both models were calculated. Baseline ANN took an average of 39.68 seconds with total parameters 63 to produce the optimum result with accuracy of 80% whereas GRU model took an average of 76.36 seconds with total parameters 195 to produce the optimum result that could correctly classify all data which is shown in the Table 4.14:

Table 4.14: Execution time of ANN and GRU model

Number of times	Time taken for ANN (seconds)	Time taken for GRU (seconds)
1	39.65199709	76.30086231
2	39.77062106	74.73838019
3	39.7531271	76.45068622
4	39.53432202	75.64199066
5	39.87325597	75.42072821
6	39.3732729	75.96102428
7	39.88473582	78.29836106
8	39.61930585	76.74690747
9	39.93815088	78.53929067
10	39.48528242	75.59035873
Average time (seconds)	39.68840711	76.36885898

More number of parameters means that model can learn more relationship between variables. Since the parameters of GRU RNN model was more than ANN model, it took more time than ANN model but achieved more accuracy and produced less loss.

4.2. Discussion

In total production data of total 1950 of 26 years data, there are 1266 deficit data, 137 normal data and 547 surplus data. From this data, we can see that there is so much poor management of food production in Nepal such that Deficit class data dominates the overall Rice production data followed by Surplus class data whereas Normal class data is very low. So accuracy of model for Normal class data is low since there is not enough Normal class data in overall data.

CHAPTER V: CONCLUSION AND FUTURE WORKS

5.1. Conclusion

GRU model has been developed for predicting the production of Rice crop in Nepal using data of 75 districts of Nepal for 26 years starting from 1991 to 2016. It has been concluded that GRU model can be used for forecasting deficit or surplus of Rice crop production in Nepal using past agricultural data. Similarly, it has also been concluded that GRU model performs better than baseline ANN model for forecasting Rice crop production.

5.2. Future Works

10 major parameters that directly affect Rice production have been used in this study. However, other micro-variables like Micro-Nutrients (Calcium, Magnesium and Sulphur), Solar Radiation and Wind Velocity which may have small effect on production can be used for further researches. Also, for hyper-tuning the parameters, instead of manual tuning, optimization algorithms such as Genetic Algorithms can be used. And furthermore, GRU models can be ensembled with Convolutional Neural Networks and other Neural Networks and compare the results with GRU model.

REFERENCES

[1] Belbase, K., & Grabowski, R. (1985). Technical Efficiency in Nepalese Agriculture. The Journal of Developing Areas.

[2] Shashtri, A., Sanjay, H., & Bhanushree, E. (2017). Prediction of Crop Yield Using Regression Techniques. International Journal of Soft Computing 12 (2): 96-102.

[3] Sellam, V., & Poovmmal, E. (October 2016). Prediction of Crop Yield using Regression Analysis. Indian Journal of Science and Technology, Vol.9 (38).

[4] Sahu, P., Mishra, P., Dhekale, B., & Padmanaban, K. (2015). Modelling and Forecasting of Area, Production, Yield and Total Seeds of Rice and Wheat in SAARC Countries and the World towards Food Security. American Journal of Applied Mathematics and Statistics Vol. 3.

[5] Bhatti, A., Ishaq, A., & Khan, A. A. (2015). Forecast Accuracy of Exponential Smoothing methods applied to four Major crops of Pakistan.

[6] Ranjeet, T., & Armstrong, L. (2014). An Artificial Neural Network for Predicting Crops Yield in Nepal. Proceedings of Asian Federation for Information Technology.

[7] Lamba, V., & Dhaka, V. (2014). Wheat Yield Prediction Using Artificial Neural Network and Crop Prediction Techniques. International Journal for Research in Applied Science and Engineering Technology.

[8] Rode, S. V., & Dahikar, S. S. (2014). Agricultural Crop Yield Prediction Using Artificial Neural Network Approach. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol. 2.

[9] Jiri Stansy, V. K. (2011). Agricultural data prediction by means of neural Network. Czech Republic: Mendel University of Brno, Zemedelska.

[10] Singh, R. K., & Prajneshu. (2008). Artificial Neural Network Methodology for Modelling and Forecasting Maize Crop Yield. Indian Agricultural Statistics Research Institute.

[11] Ji, B., Sun, Y., Yang, S., & Wan, J. (2007). Artificial neural networks for rice yield prediction in mountainous regions. Journal of Agricultural Science.

[12] Central Bureau of Statistics, GON (2017). National Population and Housing Census.

[13] Ministry of Agriculture Development, GON (2017). Statistical Information on Nepalese Agriculture.

[14] Department of Hydrology and Meteorology, GON (2016). Statistical Information on Weather Data of Nepal.

[15] Department of Food Technology and Quality Control, GON (1990). Per Capita Food Need of People in Nepal, YearBook.

[16] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014, December). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

[17] Kostadinov, S. (2017, December 16). Understanding GRU Networks. Retrieved July 14, 2018, from Towards Data Science: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be.

[18] Chollet, F. (2018). Deep Learning with Python. Shelter Island, NY 11964: Manning Publications Co.

[19] Walia, A. S. (2017, May 29). Activation functions and it's types-Which is better? Retrieved June 10, 2018, from Towards Data Science: https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f.

[20] Na, X. (2016, May 27). Why-do-many-recurrent-NNs-use-tanh. Retrieved June 4, 2018, from Quora: https://www.quora.com/Why-do-many-recurrent-NNs-use-tanh.

ANNEX A - Sample Input Data

Region	N MT/ha	P MT/ha	K MT/ha	Compost MT/ha	Maximum Temperature (0C)	Minimum Temperature (0C)	Relative Humidity (%) (8:45 am)	Relative Humidity (%) (5:45 pm)	Rainfall (mm)	Production
H	0.019607	0.001493	0.000008	0.752108	18.940000	11.980000	62.240000	44.800000	449.600000	DEFICIT
H	0.019090	0.001454	0.000008	0.732300	22.520000	16.580000	90.700000	92.620000	437.758828	DEFICIT
H	0.022111	0.001684	0.000009	0.848183	31.640000	20.900000	79.420000	76.920000	507.032067	DEFICIT
H	0.017128	0.001304	0.000007	0.657025	27.480000	18.620000	79.460000	74.600000	392.760661	DEFICIT
M	0.006069	0.000462	0.000003	2.048665	29.100000	17.320000	73.660000	62.620000	1254.400000	DEFICIT
M	0.005575	0.000425	0.000002	1.881960	27.040000	8.800000	83.640000	80.300000	1152.326108	DEFICIT
T	0.021608	0.001645	0.000009	0.142096	33.940000	22.820000	84.560000	76.840000	959.300000	SURPLUS
T	0.025811	0.001965	0.000011	0.169733	32.540000	24.060000	81.840000	72.160000	1145.880111	SURPLUS
T	0.021409	0.001630	0.000009	0.140786	33.700000	24.360000	84.040000	79.360000	950.462157	SURPLUS
H	0.062799	0.004782	0.000026	2.408939	28.220000	18.580000	80.180000	82.580000	1440.030933	SURPLUS
H	0.021221	0.001616	0.000009	0.814021	24.720000	17.280000	88.360000	85.140000	486.610647	SURPLUS
T	0.028221	0.002149	0.000012	0.185579	33.220000	24.040000	84.620000	79.100000	1252.862552	SURPLUS
H	0.019031	0.001449	0.000008	0.730028	23.060000	16.160000	85.120000	75.740000	436.400734	DEFICIT
H	0.017128	0.001304	0.000007	0.657025	26.960000	17.960000	84.740000	82.400000	392.760661	DEFICIT
T	0.021697	0.001652	0.000009	0.142676	29.940000	21.560000	82.360000	73.580000	963.216783	SURPLUS
M	0.005499	0.000419	0.000002	1.856120	31.680000	19.800000	90.440000	72.300000	1136.504538	DEFICIT
H	0.021762	0.001657	0.000009	0.834795	28.740000	20.620000	85.240000	81.560000	499.029236	DEFICIT
H	0.022920	0.001745	0.000009	0.879190	25.040000	18.220000	82.600000	88.420000	525.568019	NORMAL
T	0.019552	0.001489	0.000008	0.128575	32.560000	25.200000	80.600000	73.600000	868.024552	SURPLUS
M	0.007018	0.000534	0.000003	2.369171	23.620000	15.140000	83.620000	88.000000	1450.646138	DEFICIT

ANNEX A (contd….)

M	0.004390	0.000334	0.000002	1.481902	20.400000	8.300000	74.780000	80.700000	907.370558	DEFICIT
H	0.016274	0.001239	0.000007	0.624266	34.580000	20.920000	79.060000	60.280000	373.177745	DEFICIT
H	0.019919	0.001517	0.000008	0.764099	28.600000	20.140000	87.600000	81.620000	456.767831	DEFICIT
H	0.017661	0.001345	0.000007	0.677483	30.900000	20.320000	86.220000	77.180000	404.989933	DEFICIT
M	0.004671	0.000356	0.000002	1.576896	20.400000	8.300000	74.780000	80.700000	965.535338	DEFICIT
H	0.019227	0.001464	0.000008	0.737554	25.100000	19.340000	83.280000	90.140000	440.899711	DEFICIT
H	0.017004	0.001295	0.000007	0.652255	18.940000	10.780000	62.240000	44.800000	389.909178	DEFICIT
T	0.020003	0.001523	0.000008	0.131540	32.440000	23.260000	80.280000	84.900000	888.038273	SURPLUS
M	0.005403	0.000411	0.000002	1.823880	24.520000	11.740000	72.720000	57.640000	1116.763764	DEFICIT
T	0.021592	0.001644	0.000009	0.141985	33.220000	23.800000	83.240000	65.120000	958.553772	SURPLUS
M	0.005278	0.000402	0.000002	1.781811	27.040000	8.800000	83.640000	80.300000	1091.005076	DEFICIT
T	0.019690	0.001499	0.000008	0.129483	35.020000	21.880000	80.680000	77.440000	874.153286	SURPLUS
T	0.020776	0.001582	0.000009	0.136622	34.220000	25.200000	78.380000	69.940000	922.350961	SURPLUS
H	0.021721	0.001654	0.000009	0.833219	29.580000	20.560000	84.940000	72.560000	498.087090	DEFICIT
H	0.045487	0.003464	0.000019	1.744863	27.680000	17.980000	86.540000	76.420000	1043.055467	DEFICIT
H	0.040200	0.003061	0.000017	1.542047	31.060000	20.060000	80.100000	75.220000	921.814520	DEFICIT
H	0.016874	0.001285	0.000007	0.647280	24.720000	17.280000	88.360000	85.140000	386.935090	NORMAL
H	0.044723	0.003405	0.000019	1.715566	26.900000	18.060000	84.860000	80.880000	1025.541725	DEFICIT
H	0.019538	0.001488	0.000008	0.749479	30.340000	20.340000	87.200000	85.900000	448.028397	NORMAL
T	0.019636	0.001495	0.000008	0.129127	33.520000	25.560000	82.720000	73.140000	871.747007	SURPLUS
H	0.024939	0.001899	0.000010	0.956647	31.480000	22.580000	81.640000	84.200000	571.870305	DEFICIT

ANNEX A (contd….)

M	0.003201	0.000244	0.000001	1.080554	19.520000	10.120000	78.200000	85.860000	661.624365	DEFICIT
T	0.020386	0.001552	0.000008	0.134059	31.840000	23.740000	84.600000	74.620000	905.044959	SURPLUS
M	0.004683	0.000357	0.000002	1.580696	20.400000	8.300000	74.780000	80.700000	967.861929	DEFICIT
M	0.003397	0.000259	0.000001	1.146710	22.160000	12.020000	71.800000	65.080000	702.131980	DEFICIT
H	0.020964	0.001596	0.000009	0.804171	30.880000	20.620000	81.700000	75.880000	480.722683	DEFICIT
T	0.022875	0.001742	0.000009	0.150423	32.720000	24.400000	87.040000	79.680000	1015.521411	SURPLUS
H	0.024150	0.001839	0.000010	0.926376	29.520000	20.160000	89.780000	86.020000	553.774829	NORMAL
H	0.016891	0.001286	0.000007	0.647950	34.600000	22.260000	80.060000	75.860000	387.335569	DEFICIT
H	0.022995	0.001751	0.000010	0.882077	28.940000	19.320000	83.600000	74.260000	527.293659	DEFICIT
H	0.019178	0.001460	0.000008	0.735678	28.580000	19.660000	83.680000	78.700000	439.778449	DEFICIT
H	0.017667	0.001345	0.000007	0.677715	31.460000	19.580000	78.600000	75.280000	405.128862	DEFICIT
T	0.025152	0.001915	0.000010	0.165397	32.540000	24.060000	81.840000	72.160000	1116.612469	SURPLUS
H	0.018534	0.001411	0.000008	0.710942	32.760000	21.320000	85.480000	77.780000	424.991564	DEFICIT
H	0.017217	0.001311	0.000007	0.660431	34.600000	22.260000	80.060000	75.860000	394.796607	DEFICIT
M	0.006236	0.000475	0.000003	2.105132	23.040000	15.240000	79.060000	85.380000	1288.974999	DEFICIT
T	0.015623	0.001190	0.000006	0.102739	33.600000	24.960000	79.260000	78.000000	693.599886	SURPLUS
M	0.005412	0.000412	0.000002	1.826945	27.740000	15.200000	85.380000	87.680000	1118.640844	DEFICIT
M	0.005853	0.000446	0.000002	1.975870	32.120000	20.760000	86.520000	68.540000	1209.827411	DEFICIT
T	0.019693	0.001500	0.000008	0.129500	33.160000	24.260000	82.600000	67.840000	874.268574	SURPLUS
H	0.018717	0.001425	0.000008	0.717961	27.340000	17.980000	80.120000	81.180000	429.187499	DEFICIT
M	0.007193	0.000548	0.000003	2.428177	23.580000	16.260000	85.860000	79.080000	1486.775855	SURPLUS

ANNEX A (contd….)

T	0.018878	0.001437	0.000008	0.124142	33.520000	25.560000	82.720000	73.140000	838.095681	SURPLUS
T	0.021192	0.001614	0.000009	0.139360	32.780000	25.360000	80.960000	76.440000	940.830960	SURPLUS
H	0.024791	0.001888	0.000010	0.950993	29.300000	19.720000	89.120000	91.820000	568.490605	DEFICIT
M	0.006424	0.000489	0.000003	2.168545	24.880000	17.140000	70.520000	68.440000	1327.802933	DEFICIT
T	0.017610	0.001341	0.000007	0.115804	33.520000	25.560000	82.720000	73.140000	781.800211	SURPLUS
M	0.004390	0.000334	0.000002	1.481902	23.580000	16.260000	85.860000	79.080000	907.370558	DEFICIT
T	0.021195	0.001614	0.000009	0.139374	31.780000	23.020000	88.380000	92.600000	940.928884	SURPLUS
H	0.018208	0.001386	0.000008	0.698464	30.700000	21.540000	80.680000	68.160000	417.531966	DEFICIT
H	0.020229	0.001540	0.000008	0.775975	30.900000	20.320000	86.220000	77.180000	463.867489	DEFICIT
H	0.023072	0.001757	0.000010	0.885030	25.660000	16.940000	85.620000	84.780000	529.059059	DEFICIT
M	0.005361	0.000408	0.000002	1.809689	23.580000	16.260000	85.860000	79.080000	1108.074951	DEFICIT
H	0.021014	0.001600	0.000009	0.806091	24.720000	17.280000	88.360000	85.140000	481.870032	DEFICIT

ANNEX B – Selection of Hyper-parameters

Split loss Selection

Train/test Split	Learning Rate	Neurons	Loss and Accuracy
50:50	0.01	23	Epoch 33/1000 - 0s - loss: 0.3301 - acc: 0.9097 - val_loss: 0.6225 - val_acc: 0.8113
60:40	0.01	23	Epoch 19/1000 - 0s - loss: 0.3703 - acc: 0.8917 - val_loss: 0.5893 - val_acc: 0.8093
70:30	0.01	23	Epoch 50/1000 - 0s - loss: 0.3120 - acc: 0.8904 - val_loss: 0.5460 - val_acc: 0.8100
80:20	0.01	23	Epoch 18/1000 - 0s - loss: 0.3978 - acc: 0.8781 - val_loss: 0.5969 - val_acc: 0.7920

Hyperparameter Selection

Split	Learning Rate	Neurons	Loss and Accuracy
70:30	0.01	3	Epoch 33/1000 - 0s - loss: 0.3407 - acc: 0.9108 - val_loss: 0.5978 - val_acc: 0.8187
	0.01	5	Epoch 102/1000 - 0s - loss: 0.2836 - acc: 0.9097 - val_loss: 0.5349 - val_acc: 0.8187
	0.01	6	Epoch 42/1000 - 0s - loss: 0.3294 - acc: 0.9108 - val_loss: 0.5607 - val_acc: 0.8160
	0.01	7	Epoch 97/1000 - 0s - loss: 0.2633 - acc: 0.9118 - val_loss: 0.5462 - val_acc: 0.8107
	0.01	10	Epoch 69/1000 - 0s - loss: 0.3014 - acc: 0.9118 - val_loss: 0.5455 - val_acc: 0.8213
	0.01	15	Epoch 52/1000 - 0s - loss: 0.2960 - acc: 0.9108 - val_loss: 0.5651 - val_acc: 0.8187
	0.01	20	Epoch 65/1000 - 0s - loss: 0.2327 - acc: 0.9169 - val_loss: 0.5414 - val_acc: 0.8267
	0.01	23	Epoch 96/1000 - 0s - loss: 0.2440 - acc: 0.9149 - val_loss: 0.5165 - val_acc: 0.8160
	0.01	100	Epoch 46/1000 - 0s - loss: 0.2333 - acc: 0.9118 - val_loss: 0.5506 - val_acc: 0.8293
	0.001	23	Epoch 386/1000 - 0s - loss: 0.2612 - acc: 0.9149 - val_loss: 0.5346 - val_acc: 0.8187
	0.001	23-23	Epoch 104/1000 - 0s - loss: 0.2916 - acc: 0.9118 - val_loss: 0.5433 -

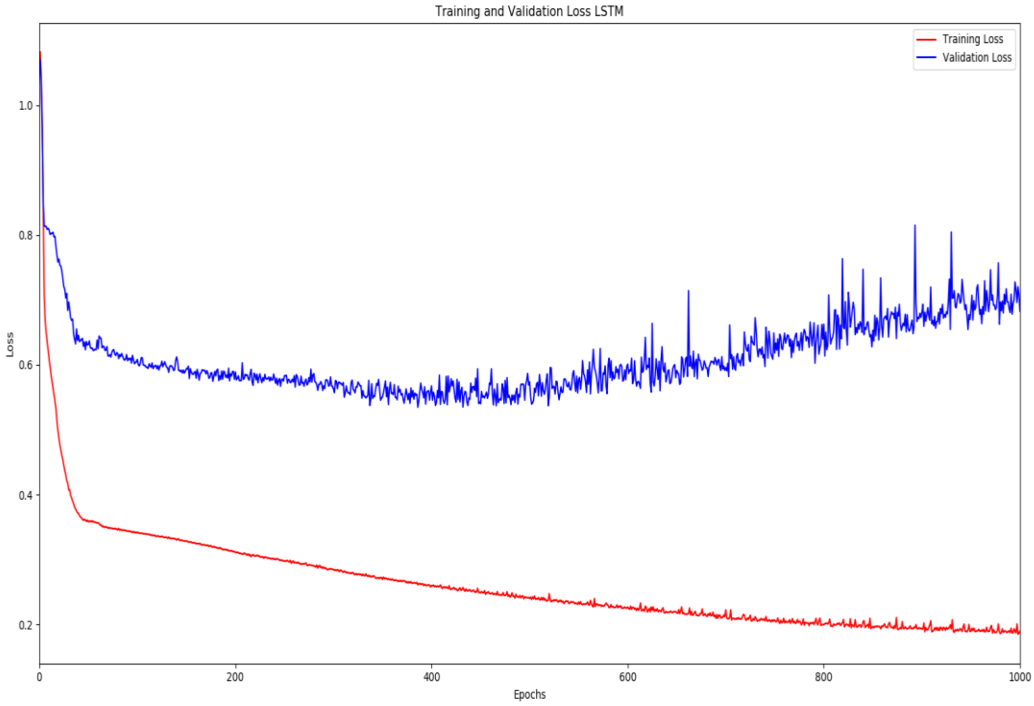
			val_acc: 0.8187
	0.001	23-23-23	Epoch 85/1000 - 0s - loss: 0.3093 - acc: 0.9128 - val_loss: 0.5933 - val_acc: 0.8187
	0.1	23	Epoch 722/1000 - 0s - loss: 0.3695 - acc: 0.9087 - val_loss: 0.5538 - val_acc: 0.8107
	0.009	23	Epoch 96/1000 - 0s - loss: 0.2649 - acc: 0.9108 - val_loss: 0.5391 - val_acc: 0.8240
	0.009	23-23	Epoch 43/1000 - 0s - loss: 0.2702 - acc: 0.9108 - val_loss: 0.5425 - val_acc: 0.8187
	0.009	23-23-23	Epoch 54/1000 - 0s - loss: 0.2438 - acc: 0.9159 - val_loss: 0.5450 - val_acc: 0.8267
	0.02	23	Epoch 27/1000 - 0s - loss: 0.2744 - acc: 0.9138 - val_loss: 0.5572 - val_acc: 0.8213
	0.02	23-23	Epoch 26/1000 - 0s - loss: 0.3181 - acc: 0.9077 - val_loss: 0.5608 - val_acc: 0.8187
	0.02	23-23-23	Epoch 712/1000 - 0s - loss: 0.2751 - acc: 0.9149 - val_loss: 0.5575 - val_acc: 0.8213
	0.01	23-23	Epoch 41/1000 - 0s - loss: 0.2756 - acc: 0.9128 - val_loss: 0.5633 - val_acc: 0.8187
	0.01	23-23-23	Epoch 15/1000 - 0s - loss: 0.3244 - acc: 0.9118 - val_loss: 0.5657 - val_acc: 0.8187
	0.01	23-3	Epoch 97/1000 - 0s - loss: 0.2729 - acc: 0.9087 - val_loss: 0.5237 - val_acc: 0.8267
	0.01	23-10	Epoch 65/1000 - 0s - loss: 0.2496 - acc: 0.9108 - val_loss: 0.5560 - val_acc: 0.8187
	0.01	23-30	Epoch 65/1000 - 0s - loss: 0.2191 - acc: 0.9241 - val_loss: 0.5630 - val_acc: 0.8107

ANNEX C - Loss-Epoch Curves

Learning Rate =0.001, Neurons=23

Epoch 386/1000

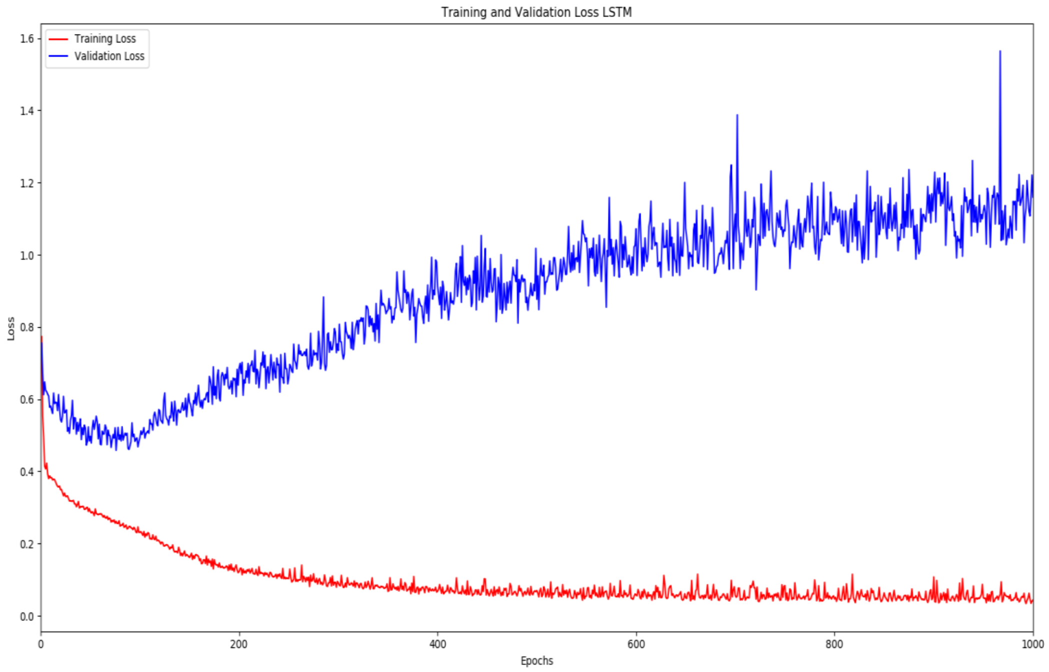
- 0s - loss: 0.2612 - acc: 0.9149 - val_loss: 0.5346 - val_acc: 0.8187



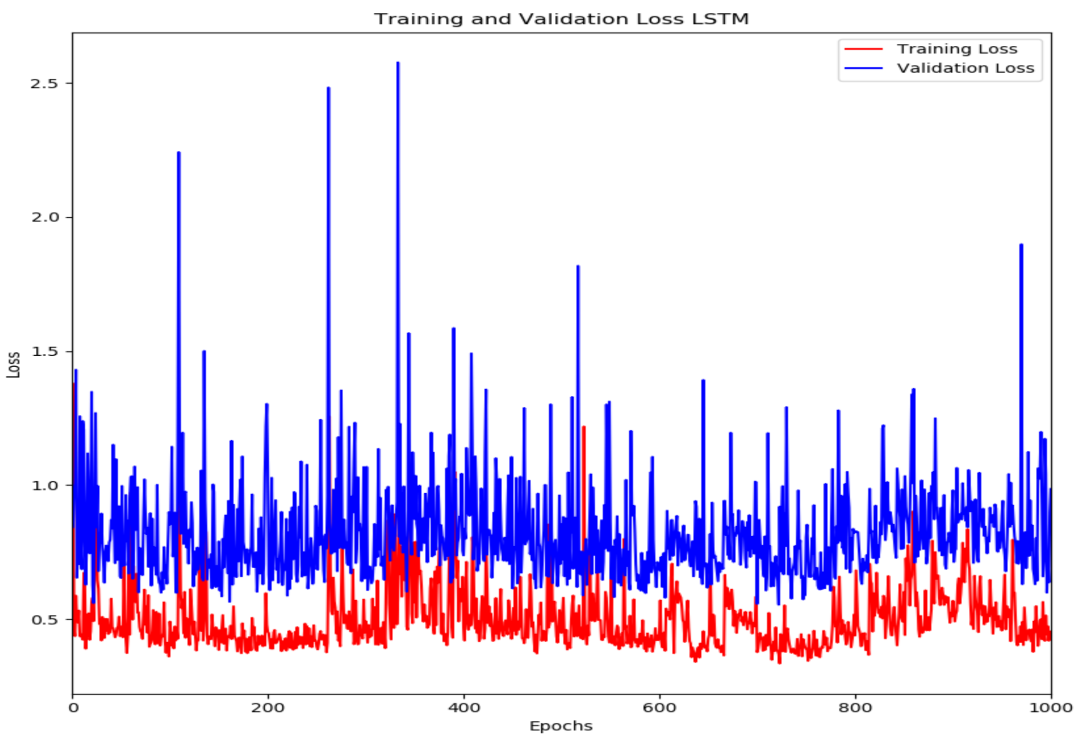
Learning Rate =0.01 neuron=23

Epoch 96/1000

- 0s - loss: 0.2440 - acc: 0.9149 - val_loss: 0.5165 - val_acc: 0.8160

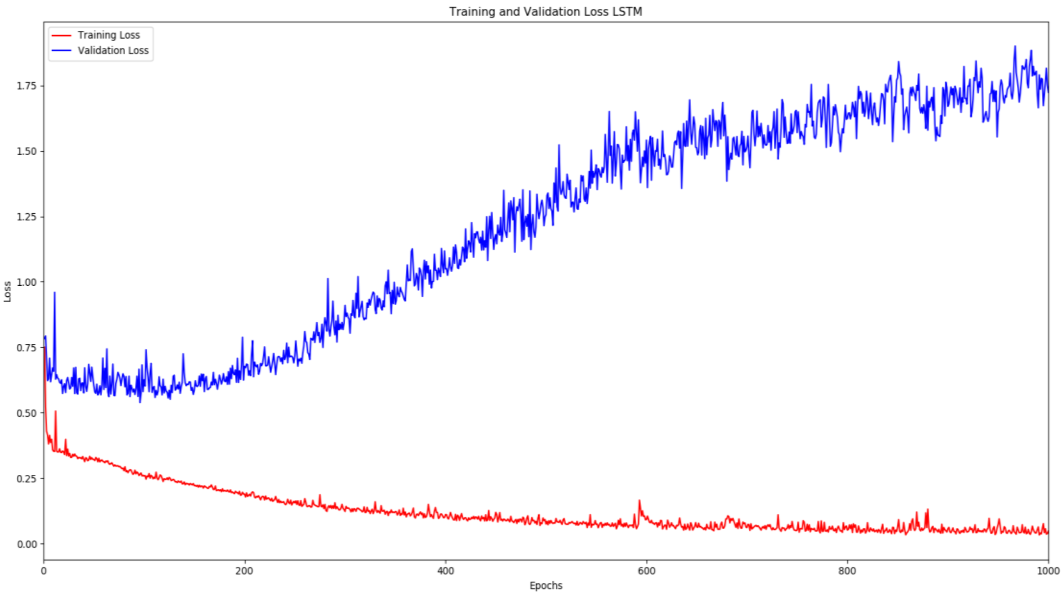


lr=0.1 neuron=23
Epoch 722/1000
- 0s - loss: 0.3695 - acc: 0.9087 - val_loss: 0.5538 - val_acc: 0.8107



lr=0.009 neuron=23
Epoch 96/1000

- 0s - loss: 0.2649 - acc: 0.9108 - val_loss: 0.5391 - val_acc: 0.8240



ANNEX D – Sample Output data

Sample output data for year 2016

District	Actual Production	Predicted Production
Achham2016	DEFICIT	DEFICIT
Arghakhanchi	DEFICIT	DEFICIT
Baglung	DEFICIT	DEFICIT
Baitadi	DEFICIT	DEFICIT
Bajhang	DEFICIT	DEFICIT
Bajura	DEFICIT	DEFICIT
Banke	SURPLUS	SURPLUS
Bara	SURPLUS	SURPLUS
Bardiya	SURPLUS	SURPLUS
Bhaktapur	DEFICIT	DEFICIT
Bhojpur	SURPLUS	SURPLUS
Chitwan	NORMAL	NORMAL
Dadeldhura	DEFICIT	SURPLUS
Dailekh	DEFICIT	DEFICIT
Dang Deokhuri	SURPLUS	SURPLUS
Darchula	DEFICIT	DEFICIT
Dhading	DEFICIT	DEFICIT
Dhankuta	DEFICIT	DEFICIT
Dhanusha	NORMAL	SURPLUS
Dolakha	DEFICIT	DEFICIT
Dolpa	DEFICIT	DEFICIT
Doti	DEFICIT	DEFICIT
Gorkha	DEFICIT	DEFICIT
Gulmi	DEFICIT	DEFICIT
Humla	DEFICIT	NORMAL
Ilam	DEFICIT	DEFICIT
Jajarkot	DEFICIT	DEFICIT
Jhapa	SURPLUS	SURPLUS
Jumla	DEFICIT	DEFICIT
Kailali	SURPLUS	SURPLUS
Kalikot	DEFICIT	DEFICIT
Kanchanpur	SURPLUS	SURPLUS
Kapilvastu	SURPLUS	SURPLUS
Kaski	DEFICIT	DEFICIT
Kathmandu	DEFICIT	DEFICIT
Kavre	DEFICIT	DEFICIT
Khotang	DEFICIT	DEFICIT
Lalitpur	DEFICIT	DEFICIT

ANNEX –D (contd...)

District	Actual Production	Predicted Production
Lamjung	SURPLUS	DEFICIT
Mahottari	DEFICIT	SURPLUS
Makwanpur	DEFICIT	DEFICIT
Manang	DEFICIT	DEFICIT
Morang	SURPLUS	SURPLUS
Mugu	DEFICIT	DEFICIT
Mustang	DEFICIT	DEFICIT
Myagdi	DEFICIT	DEFICIT
Nawalparasi	SURPLUS	SURPLUS
Nuwakot	SURPLUS	DEFICIT
Okhaldhunga	DEFICIT	DEFICIT
Palpa	DEFICIT	DEFICIT
Panchthar	DEFICIT	DEFICIT
Parbat	NORMAL	DEFICIT
Parsa	SURPLUS	SURPLUS
Pyuthan	DEFICIT	DEFICIT
Ramechhap	DEFICIT	DEFICIT
Rasuwa	DEFICIT	DEFICIT
Rautahat	DEFICIT	SURPLUS
Rolpa	DEFICIT	DEFICIT
Rukum	DEFICIT	DEFICIT
Rupandehi	SURPLUS	SURPLUS
Salyan	DEFICIT	DEFICIT
Sankhuwasabha	SURPLUS	DEFICIT
Saptari	DEFICIT	SURPLUS
Sarlahi	DEFICIT	SURPLUS
Sindhuli	DEFICIT	DEFICIT
Sindhupalchok	DEFICIT	DEFICIT
Siraha	DEFICIT	SURPLUS
Solukhumbu	DEFICIT	DEFICIT
Sunsari	SURPLUS	SURPLUS
Surkhet	DEFICIT	DEFICIT
Syangja	SURPLUS	DEFICIT
Tanahu	DEFICIT	DEFICIT
Taplejung	DEFICIT	DEFICIT
Terhathum	DEFICIT	SURPLUS
Udayapur	DEFICIT	DEFICIT

ANNEX –D (contd...)

Sample Output data for year 2017

District	Actual Production	Predicted Production
Achham2017	DEFICIT	DEFICIT
Arghakhanchi	DEFICIT	DEFICIT
Baglung	DEFICIT	DEFICIT
Baitadi	DEFICIT	DEFICIT
Bajhang	DEFICIT	DEFICIT
Bajura	DEFICIT	DEFICIT
Banke	DEFICIT	SURPLUS
Bara	SURPLUS	SURPLUS
Bardiya	SURPLUS	SURPLUS
Bhaktapur	DEFICIT	NORMAL
Bhojpur	NORMAL	DEFICIT
Chitwan	DEFICIT	SURPLUS
Dadeldhura	DEFICIT	DEFICIT
Dailekh	DEFICIT	DEFICIT
Dang Deokhuri	NORMAL	NORMAL
Darchula	DEFICIT	DEFICIT
Dhading	DEFICIT	DEFICIT
Dhankuta	DEFICIT	DEFICIT
Dhanusha	SURPLUS	SURPLUS
Dolakha	DEFICIT	NORMAL
Dolpa	DEFICIT	DEFICIT
Doti	DEFICIT	DEFICIT
Gorkha	DEFICIT	DEFICIT
Gulmi	DEFICIT	DEFICIT
Humla	DEFICIT	DEFICIT
Ilam	DEFICIT	DEFICIT
Jajarkot	DEFICIT	DEFICIT
Jhapa	SURPLUS	SURPLUS
Jumla	DEFICIT	DEFICIT
Kailali	SURPLUS	SURPLUS
Kalikot	DEFICIT	DEFICIT
Kanchanpur	SURPLUS	SURPLUS
Kapilvastu	SURPLUS	SURPLUS
Kaski	DEFICIT	DEFICIT
Kathmandu	DEFICIT	SURPLUS
Kavre	DEFICIT	DEFICIT
Khotang	DEFICIT	DEFICIT
Lalitpur	DEFICIT	DEFICIT

ANNEX –D (contd...)

District	Actual Production	Predicted Production
Lamjung	SURPLUS	DEFICIT
Mahottari	DEFICIT	SURPLUS
Makwanpur	DEFICIT	DEFICIT
Manang	DEFICIT	DEFICIT
Morang	SURPLUS	SURPLUS
Mugu	DEFICIT	DEFICIT
Mustang	DEFICIT	DEFICIT
Myagdi	DEFICIT	DEFICIT
Nawalparasi	SURPLUS	SURPLUS
Nuwakot	SURPLUS	DEFICIT
Okhaldhunga	DEFICIT	DEFICIT
Palpa	DEFICIT	DEFICIT
Panchthar	DEFICIT	DEFICIT
Parbat	DEFICIT	DEFICIT
Parsa	SURPLUS	SURPLUS
Pyuthan	DEFICIT	DEFICIT
Ramechap	DEFICIT	DEFICIT
Rasuwa	DEFICIT	DEFICIT
Rautahat	DEFICIT	SURPLUS
Rolpa	DEFICIT	SURPLUS
Rukum	DEFICIT	SURPLUS
Rupandehi	SURPLUS	SURPLUS
Salyan	DEFICIT	DEFICIT
Sankhuwashava	NORMAL	DEFICIT
Saptari	NORMAL	SURPLUS
Sarlahi	DEFICIT	SURPLUS
Sindhuli	DEFICIT	DEFICIT
Sindhupalchok	DEFICIT	SURPLUS
Siraha	NORMAL	SURPLUS
Solukhumbu	DEFICIT	DEFICIT
Sunsari	NORMAL	SURPLUS
Surkhet	DEFICIT	SURPLUS
Syangja	NORMAL	DEFICIT
Tanahu	DEFICIT	DEFICIT
Taplejung	DEFICIT	DEFICIT
Terhathum	DEFICIT	DEFICIT
Udayapur	DEFICIT	DEFICIT