EFFECTS OF DATA PREPROCESSING ON THE PREDICTION ACCURACY OF ARTIFICIAL NEURAL NETWORK MODEL IN HYDROLOGIC TIME SERIES

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

By

ANIRUDDHA GOPAL BANHATTI
(Register Number: 082012AM08P05)

DEPARTMENT OF APPLIED MECHANICS AND HYDRAULICS
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE – 575 025

September, 2012

D E C L A R A T I O N

by the Ph.D. Research Scholar

I hereby *declare* that the Research Thesis entitled **"Effect Of Data Preprocessing On The Prediction Accuracy Of Artificial Neural Network Model in Hydrologic Time Series"** Which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Civil Engineering** is a *bonafide report of the research work carried out by me*. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

ANIRUDDHA GOPAL BANHATTI
(Register Number: 082012AM08P05)
Department of Applied Mechanics and Hydraulics

Place: NITK-Surathkal
Date: 18 - 09 - 2012

C E R T I F I C A T E

This is to certify that the Research Thesis entitled **"Effect of Data Preprocessing on The Prediction Accuracy of Artificial Neural Network Model in Hydrologic Time Series"** submitted by **Aniruddha Gopal Banhatti** (Register Number: **082012AM08P05**) as the record of the research work carried out by him, is accepted as the Research Thesis submission in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. Paresh Chandra Deka
Research Guide
(Name and Signature with Date and Seal)

Prof. M. K. Nagaraj
Chairman – DRPC (Name and Signature with Date and Seal)

DEDICATED

TO

RADHIKA, ANUPAM

AND

MANJU

ACKNOWLEDGEMENT

It is my pleasure to express profound gratitude and indebtedness towards my research supervisor **Dr. Paresh Chandra Deka**, Associate professor, Department of Applied Mechanics and Hydraulics, for his continued inspiration, motivation, support, discussions, and great patience throughout this research, which made this study possible. It is a valuable experience to learn many aspects from him as a good teacher. I admire among his other qualities, kindness and balanced approach towards success and failure; his scientific foresight and excellent knowledge have been crucial to the accomplishment of this work; who managed nicely to spare valuable time for guidance, valuable suggestions and excellent supervision of my research work. I consider myself privileged for having had the opportunity to conduct research in the area of soft computing techniques under his able supervision.

I am greatly indebted to Research Progress Appraisal Committee members , Prof. Rammohan Reddy, Department of Information Technology and Professor Subhash Yaragal, Department of Civil Engineering, for their critical evaluation and constructive comments and valuable suggestions during the progress of the work helped me to improve the quality of work.

I also extend my heartfelt thanks to Prof. M. K. Nagaraj, Prof. SubbaRao, Head, Department of Applied Mechanics and Hydraulics and Chairman RPAC for his continuous support, encouragement, and timely help, also for providing me all the necessary departmental facilities during my research period.

I gratefully acknowledge Prof. Mayya S.G. and Prof. Lakshman Nandagiri, Department of Applied Mechanics and Hydraulics, for their continuous support, care, timely help and their good wishes during the course of my work.

I am also grateful to all other faculty members, Department of Applied Mechanics and Hydraulics, NITK, Surathkal, for helping me directly or indirectly during my stay and research work.

I take this opportunity to express thanks to my friends Sreenivasulu, D., Latifa Haque, and Leeladhar Pammar, (Research Scholars) and Sujay Raghavendra N for rendering my stay in the NITK Campus more than wonderful.

I also acknowledge the help and support provided non teaching staff, Sri. Jagadish B Foreman, Sri Balakrishna, Sri. Ananda Devadiga, Sri. Gopalakrishna, Sri. Padmanabha Achary, Mr. Harish Salians, Mr. Harish D and Mrs. Prathima Prakash for their support and help during the research work.

The inspiration and support given by the other fellow Research Scholars of the Department of Applied Mechanics and Hydraulics have also been much appreciated.

Without the support, patience and encouragement from my wife Manju and my daughter Radhika I could never have been able to submit this work. Finally, I would like to thank the Almighty God for blessing me with good health, ability to work hard and guiding me to every success in life.

ANIRUDDHA GOPAL BANHATTI

Place: NITK, Surathkal
Date: 18-09-2012

ABSTRACT

The accurate prediction of hydrological behavior in both urban and rural watershed can provide valuable information for the urban planning, land use, design of civil projects and water resources management. Hydrology system is influenced by many factors such as weather, land cover, infiltration, evapotranspiration, so it includes a good deal of stochastic dependent component, multi-time scale and highly non-linear characteristics. Hydrologic time series are often non-linear and non- stationary. In spite of high flexibility of Artificial Neural Network (ANN) in modeling hydrologic time series, sometimes signals are highly non-stationary and exhibit seasonal irregularity. In such situation, ANN may not be able to cope with non-stationary data if pre-processing of input and/or output data is not performed. Pre-processing data refers to analyzing and transforming input and output variables in order to detect trends, minimize noise, underline important relationship and flatten the variables distribution in a time series. These analyses and transformations help the model learn relevant patterns. Pre-processing techniques, which facilitate stabilization of the mean and variance, and seasonality removal, are often applied to remove non- stationary aspect in data used to build soft computing models.

In this study, different data pre-processing techniques are presented to deal with irregularity components that exist in a hydrologic time series data of the Brahmaputra basin within India at the Pandu gauging station near Guwahati city and Pancharatna gauging station further 150km downstream of Pandu by using daily time unit and their properties are evaluated by performing one step ahead flow forecasting using ANN. Three different preprocessed datasets are used for the analysis. Various ANN models are generated by varying network internal architecture with different input scenarios.

The model results were evaluated by using Root Mean Square Error (RMSE)and Mean Absolute Percentage Error (MAPE) and found that Logarithmic based pre-processing techniques provide better forecasting performance among various pre-processing techniques.

The results indicate that detecting non-stationary aspect and selecting an appropriate pre-processing technique is highly beneficial in improving the prediction performance of ANN model.

Keywords: Brahmaputra River, Gauging Station, Pandu, Pancharatna, Guwahati, Time Series, Data Preprocessing, ANN, FFBP, Activation Function, RMSE, MAPE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES –CONTD

LIST OF FIGURES – CONTD.

LIST OF FIGURES – CONTD.

LIST OF TABLES Title

NOMENCLATURE Symbol Description

ANN	Artificial Neural Network
NN	Neural Network
FFBP	Feed-forward back Propagation
LM	Levenberg-Marquartz
MLP	Multilayer Perception
RBFNN	Radial Basis Function Neural Network
GRNN	Generalized Regression Neural Network
SOM	Self Organized Map
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error CC
	Coefficient of Correlation
CE	Coefficient of Efficiency
MAPE	Mean Absolute Error MSE
	Mean Square Error
GWL	Groundwater level

CHAPTER - 1

INTRODUCTION

1.1 Introduction

According to the Mc Cuen definition of Hydrology (1997), it is the scientific study of water and its properties, distribution and effects on the earth's surface, soil and atmosphere. Thus the definition being all encompassing, most hydrological processes have an extremely complex nature affected by myriad of local as well as global factors ranging from the shape of leaf of predominant vegetation in a catchment to the atmospheric data correlation via satellites. Scientists have gone to ridiculous extremes in the name of studying every single factor that affects precipitation, rainfall-runoff correlation for the given catchment, various sources that contribute to the stream flow and then predicting the stream flow. Despite this, we still have not achieved acceptable quality in conceptual modeling.

In recent decades, with the advent of data forecasting using ANNs all these factors can be treated as extraneous and the prediction can be based entirely on the previous data available of a single variable for which prediction is desired.

The conceptual models fall broadly under 3 categories:

Empirical, Geomorphology based and Physics based.

Empirical models treat hydrologic systems e.g. a watershed, as a black box and try to establish a relationship between inputs such as rainfall, temperature, humidity, stem flow, vegetation cover etc. and outputs such as stream flow measured at a gauging station at the egress of a catchment. Lumped models fall under this category. (Blackie and Eeles 1975)

Geomorphology based models represent an improvement over the empirical models. These models simulate the watershed and the stream network quite well but the large number of assumptions for simplification of complex natural phenomena affects the quality of these models. (e.g. Gupta and Waymire 1983; Corradini and Singh 1985)

Physically based models try to reduce all natural phenomena to the physics involved and reduce everything to a set of complex differential equations involving a high number of variables (Freeze and Harlan 1969). Measurement of these variables in an accurate manner and reliable recording of the same is the prime requirement for these models. This kind of data is rarely available even in research watersheds which are heavily instrumented at very high costs.

Therefore the data driven approach of ANN becomes an attractive alternative and improving the prediction capacities for a single variable of a time series is the need of the hour.

Time series forecasting has received tremendous attention of researchers in the last few decades. This is because the future values of a physical variable, which are measured in time at discrete or continuous basis, are needed in efficient planning, design and management activities (Jain and Kumar, 2007). Conventional time series models such as Box-Jenkins methods of autoregressive (AR), Autoregressive Moving Average (ARMA), Autoregressive Moving Average with Exogenous inputs (ARMAX) etc., have been used by researchers since long back. However, these models suffer in the accuracy and applicability aspects due to the certain assumptions. A common assumption in the time series analysis is that time series data have constant mean and variance, i.e. they are stationary. This is normally true except when shocks are administered to the system generating the series, resulting in non-stationary values in variance, or there is a trend in the series, resulting in non-stationary nature in the mean.

Use of ANN technique has been increased for the past few decades in surface water hydrology for the purpose of forecasting, modelling, and many more problems. A lot of successful applications have shown that ANN Provides powerful deterministic tool for time series modelling (Zhang et al., 1998; Nag and Mitra, 2002). Comparisons were made between traditional methods and ANN on time series forecasting (Hamid and Zahid, 2004). The supports for ANN in time series analysis are the capability of non-linear modelling in real world complex phenomena. Also, ANN is a non-parametric method and prior knowledge is not mandatory. All these features make ANN attractive for time series modelling and forecasting.

Recently, ANN has shown great ability in modelling and forecasting nonlinear hydrologic time series (Deka and Chandramoulli, 2005; Sreenivasulu and Deka, 2011). Although classic time series models like autoregressive moving average (ARMA) are widely used for hydrologic time series forecasting, they are based on linear models assuming the data are stationary and have limited ability to capture non-stationarities and non-linearities in hydrologic data. ANNs are found suitable for handling huge amounts of dynamic, nonlinear and noisy data when underlying physical relationships are not fully understood. ANN has found increasing considerations in forecasting theory, leading to successful applications in various forecasting domains. ANN can learn from examples (past data), recognize a hidden pattern in historical observations and use them to forecast future values. In addition to that, they are able to deal with incomplete information or noisy data and can be very effective especially in situations where it is not possible to define the rules, relationships or steps that lead to the solution of a problem.

The attractive features of ANN to various forecasting domains are many. Being a data driven learning machine as opposed to conventional model based approaches, permitting universal approximation of arbitrary linear or non-linear functions, and therefore offering great flexibility in learning, the generator of noisy data from examples and generalizing structure from it without priori assumptions (Zhang et al.1998). Due to their flexibility, neural network lacks systematic procedure for model development. Therefore obtaining a reliable neural network model involves selecting a large number of parameters experimentally through trial and error (Kaastra and Boyed, 1996).

Despite many satisfactory characteristics (Zhang et al, 1998) of ANNs, developing an ANN model for a particular forecasting problem is a non-trivial task. Several authors such as Plummer (2000), Xu and Chen (2001), Lam (2004) have provided an insight on issues in developing ANN model for forecasting. These modelling issues must be considered carefully because it may affect the performance of ANNs. Based on their studies, some of the discussed modelling issues in constructing ANN forecasting model are the selection of network architecture, learning parameters and data pre-processing techniques as applied to the time series data.

In spite of high flexibility of ANN in modelling hydrologic time series, sometimes signals are highly nonstationary and exhibit seasonal irregularity. In such situation, ANN may not be able to cope with non-stationary data if preprocessing of input and/or output data is not performed (Cannas et al., 2006). Simple ANN systems as well as complicated hybrid systems have been used to analyze real world time series which are usually characterized by mean and variance changes, seasonality and other local behavior. Such real world time series are not only invariably non-linear and non-stationary, but also incorporate significant distortions due to both 'knowing and unknowing misreporting' and 'dramatic changes in variance' (Granger'94). The presence of these characteristics in time series stress desirability of data pre-processing (Nelson et al. 1999; Zhang et al. 2001; Zhang and Qi, 2005; Deka and Prahlada, 2012). These studies have focused on investigating the ability of NN to model non-stationary time series and effect of data pre-processing on forecast performance of NN.

Preprocessing techniques, which facilitate stabilization of the mean and variance, and seasonality removal, are often applied to remove non-stationarity in data used to build soft computing models.

Data pre-processing is an important step in developing ANN application, which could affect model accuracy and results. Preprocessing data refers to analyzing and transforming input and output variables in order to detect trends, minimize noise, underline important relationship and flatten the variables' distribution. These analysis and transformations help the model learn relevant patterns. Before data is used by an algorithm, it must go through several transformations in order to prepare the input data. The success of an algorithm greatly depends on the quality of input data. As different methods can handle only different samples, it is recommended to exploit certain data features with the purpose of finding out which pre-processing transformation works best.

1.2 Problem background

The performance of ANN in forecasting is influenced by ANN modelling, that is the selection of the most relevant network architecture and network design. Poor selection of parameter settings can lead to slow convergence and incorrect output (Kong and Martin, 1995). One critical decision is to determine the appropriate network architecture, that is, number of layers, the number of nodes in this layer, and the number of arcs which interconnect with the nodes. The network design decisions include the selection of activation function in the hidden and output neurons, the training algorithm, data transformation or normalization method, training and test set, and performance measures. Zhang (1992), Kong and Martin (1995), had focused their studies on parametric effects on building a BP (Back propagation) network for a particular forecasting problem. The issues on modelling fully connected feed forward networks for forecasting had been discussed by Zhang et al (1998). Maier and Dandy (2000) have reviewed the modelling issues and outlined the steps that should be followed in developing ANN model for predicting and forecasting water resources variables.

When BP algorithm was introduced in 1986, there has been much development in the use of ANN for forecasting by a number of researchers such as Zhang (1992), Kong and Martin (1995), Lopes et al(2000), Crone et al(2004), Deka and Chandramouli (2005). Thus, this study uses a Back propagation network to predict the hydrologic behavior of the Brahmaputra basin within India at the Pandu and Pancharatna gauging stations located on the main stem of Brahmaputra by using daily flow time series data. In particular, due to high data non-stationarity and seasonal irregularity, typical of a Himalayan weather regime, the role of data pre-processing through logarithmic transformation and detrending transformation has been investigated. This study examines the effect of network parameters through trial and error by varying network structures based on the number of input nodes, activation functions and data pre-processing in designing BP network forecasting model.

The novelty in this work is that, unlike other schemes reported in the literature, our method explicitly takes the statistical properties of the time series into consideration, and only recommends Log-based pre-processing when the properties of the data indicate that such pre-processing is appropriate. If a sophisticated method is used without understanding the underlying properties of the time series, then ironically for certain classes of time series, the forecast are worse than by simpler methods.

1.2 Study Area

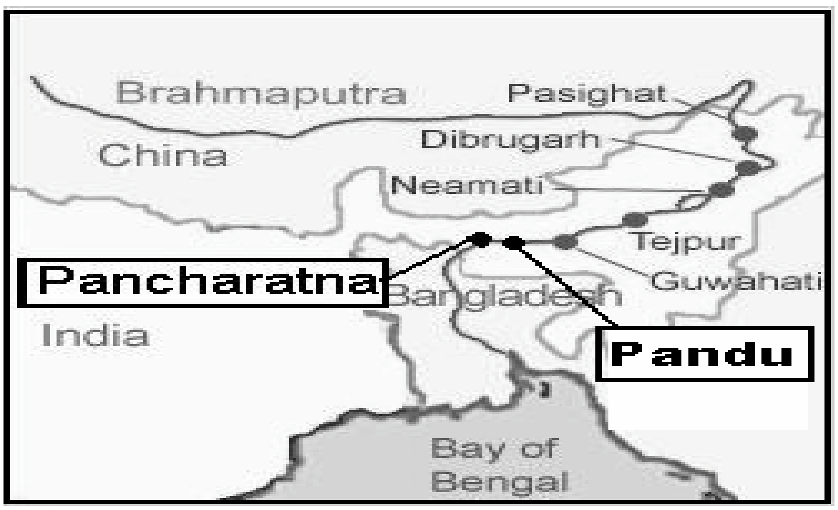


Fig. 1.1 Map of study area

The study area is located in the Brahmaputra main stream. The two discharge gauging sites namely Pandu and Pancharatna are selected for the study as these two stations contribute heavy discharge flows to the main-stream of the Brahmaputra basin causing frequent floods to the downstream area. The predictions are carried out using the discharge records of these two stations.

1.3.1Gauging stations

Brahmaputra River within India was selected for the study. The Brahmaputra originates in Tibet region in China is the fourth largest river in the world in terms of average discharge at mouth, with a flow of 19,830 cumecs (Goswami, 1985). The hydrologic regime of the river responds to the seasonal rhythm of the monsoons and to the freeze-thaw cycle of the Himalayan snow. The discharge is highly fluctuating in nature. Discharge per unit drainage area in the Brahmaputra River is among the highest of major rivers of the world. The basin lies between latitudes $24^0 13'$ and $31^0 30'9$ North and longitudes 82^0 and $96^0 49'$ East**.** The total catchment area is 5, 80,000 sq.km covering the full length from the source to the confluence of Bay of Bengal. The average width of the river is 5.46 km. The average rainfall in the catchment is 2500 mm. The catchment area up to Pandu station is 500000 sq.km and up to Pancharatna it is 532,000 sq.km. The annual rainfall in the Assam part is 2300mm with annual Evapotranspiration 1230 mm while annual runoff is 1251 with runoff coefficient 0.54. The distance between both the stations is 150 km. More than 100 tributaries join in the main stem of Brahmaputra River within Assam State (Sharma, 2005). The location of the two discharge gauging stations namely Pandu and Pancharatna are shown in the figure 1.1 below.

Large variations of discharge within a short span of time are noticed during the flood season, with maximum difference of about 17000 cumecs in 24hours (June 7-8, 1990) and 24000 cumecs in 48 hours (June 7-9, 1990) being recorded in rising limb. Maximum reduction of flow on recession limb was 12000 cumecs over 24 hours (Sept 21-22, 1977). Most of hydrographs exhibits multiple flood peaks occurring at different times from June to October (Sharma,2005). At Pancharatna, the Brahmaputra yields 0.0509 cumecs per sq.km and the mean annual flood discharge is 51,156 cumecs. Assam thus account for 9.4% of the total flood prone area in India mainly because of Brahmaputra River and its tributaries. The main factors causing extensive floods are the adverse physiography of the region, heavy rainfall, excessive sediment due to frequent earthquakes and hill slides, reduction of forest coverage and encroachment of the riverine area due to population explosion. The stations are located in the floodplain of the river where almost every year flood damage exceeds around 0.10 billion dollar and renders the people homeless.

The annual records show that the flow of the river causes devastating effect both in terms of loss of life and property. Hence, it is important to study the flow characteristics with efficient forecasting which can enhance decision support system for proper management strategy.

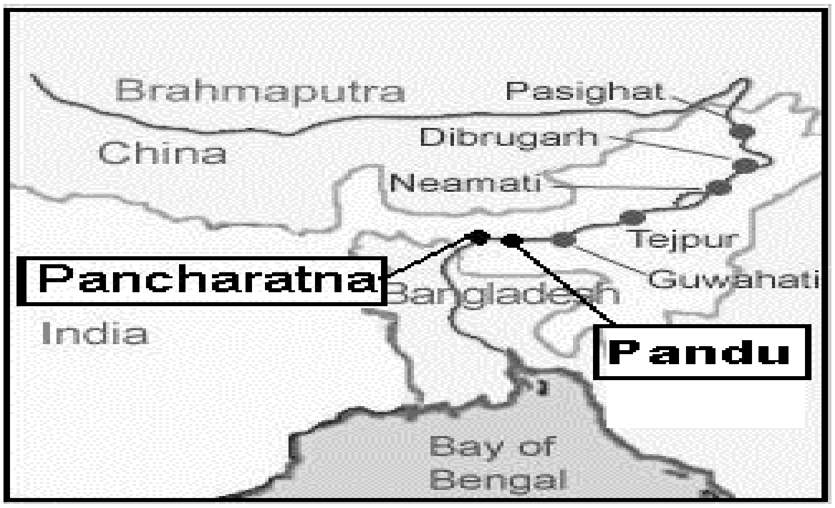


Figure.1.1 Study Area

1.4 Problem statement

The problem statement of this study may be stated as follows:

How does the selection of the parameters in network modelling namely number of input nodes, activation functions, number of neurons and data preprocessing techniques affect the forecasting capability of ANN in time series hydrologic forecasting?

1.5 Research Objectives

Considering the problem background, it is proposed to investigate whether the forecasting performance of ANNs improves by using data preprocessing techniques previously not tried with hydrological time series comparing with unprocessed raw data.

This research attempts to explore:

- The effectiveness of data preprocessing technique on ANN modelling and forecasting performance.
- The generalization capability of the ANN by varying the network structure.

1.6 Specific Objectives

- Development of various ANN model for river flow forecasting using raw and preprocessed daily time series data for multiple input scenarios.
- To assess the performance of model for two different gauging stations carrying different statistical properties for one day lead-time forecasting.
- Selection of best network for river flow predictive model.

1.7 Scope of study

The scope of this study is as follows:

1. Real hydrological time flow forecasting using neural network series flow data obtained from Water resources department, Govt. of Assam, India from January1980 to December1999 are used as input to the ANN model.
2. The MLP network with three layers (one hidden, an input and an output layer) is used.
3. Trial and error design procedures are employed to arrive at an acceptable structure and parameters namely: data preprocessing techniques, number of input nodes, number of hidden neurons and activation function of ANN model.
4. Different data preprocessing techniques are presented to deal with irregularity components exist in time series data and their properties are evaluated by performing one step ahead flow forecasting using neural network.
5. The network hidden nodes are varied from 1 to 10 nodes to see its effect onto the network while the number of input nodes varies based on lagged variables.
6. The output of the network is the forecast of one step ahead (one day ahead) flow.
7. Activation functions LOGSIG (Logistic sigmoid function), TANSIG (Non – linear Sigmoidal function) and PURELIN (linear function) are used.

8. Rainfall, infiltration, evapotranspiration and other outside factors are not considered and included in the estimation.

1.8 Organization of the Thesis

This thesis comprises of five chapters as follows.

Chapter 1 Introduction presents the relevant information pertaining to time series and further deals with the problem identification, study area and its significance, research objectives, assumptions and limitations of research, overview of the conceptual basis for the research.

Chapter 2 Literature Review discusses the time series modelling, conventional methods of forecasting and thereby explains the effects of data preprocessing on model accuracy of ANN.

Chapter 3 Materials and Methodology describes the different datasets used and explains the methodology adopted in order to achieve the research objectives. This includes the essential background information, a description of the structure and terminology of various ANN models.

Chapter 4 Results and Discussion describes the method of evaluation and goes on to present the analysis of the results obtained from the developed models and network performance for different input configuration.

Chapter 5 Summary and Conclusions presents summary of research work carried out, contribution and conclusions. Further, the limitations of the research work and scope for future work are included towards the end.

CHAPTER – 2

LITERATURE REVIEW

2.1 Introduction

The present chapter focuses on a review of research carried out in the past involving the time series analysis, effect of data preprocessing and suitability of Artificial Neural Networks algorithms for river flow forecasting.

It is attempted to make the literature review on the applications of ANN in water resources engineering particularly in the following categories.

2.2 Causes of Streamflow Changes

The streamflow is one of the important hydrologic variable at a location in a riverin a catchment usually measured on daily,weekly or monthly basis.The accurate streamflow forecasts at a particular location is very much important in water resources management and design activities such as flood control structure,bridges,irrigation structures.The streamflow process in a catchment is complex and nonlinear affected by many physical factors like intensity and duration of rainfall events,catchment characteristics, geomorphological and climatic characteristics.The influence of these factors and their combination in generating streamflow is an extremely complex physical process and is not understood clearly (Zhang and Govindaraju,2000) ANNs have been proposed as efficient tools under these situations which can reproduce the unknown relationship between a set of explanatory variables and output variables (K.Chakravarty et al,1992).

2.3ANN Applications in Hydrology

In recent decades, considerable interest has been raised for various ANN algorithms over their practical applications, because the neural networks can automatically develop a forecasting model through a simple process of the historic data. Such a training process enables the neural system to capture the complex and non-linear relationships that are not easily analyzed by conventional methods (Lin and Chen 2004). ANNs were first developed in 1940s around more than 60 years ago. Since then, it has been widely used on pattern/speech recognition and image/signal processing in the field of science and technology (Widrow and Lehr, 1992). The application of ANN in hydrology started in the early 1990s (ASCE, 2000a; 2000b).

Artificial neural networks are powerful tools that can learn to solve problems in a way similar to the human brain. An artificial neuron is a computational model inspired in the natural neurons. The natural neurons receive signals through *synapses* located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain *threshold*), the neuron is *activated* and emits a signal though the *axon*. This signal might be sent to another synapse, and might activate other neurons. The complexity of real neurons is highly abstracted when modeling with artificial neurons. These basically consist of *inputs* (like synapses), which are multiplied by *weights* (strength of the respective signals), and then computed by a mathematical functionwhich determines the *activation* of the neuron. Another function (which may be the identity) computes the *output* of the artificial neuron (sometimes in dependence of a certain *threshold*). ANNs combine artificial neurons in order to process information. The schematic biological neuron and artificial neuron are shown in Figure2.1 and Figure 2.2. Figure 2.2 shows the structure of the simple ANN. It is a combination of many single neurons.

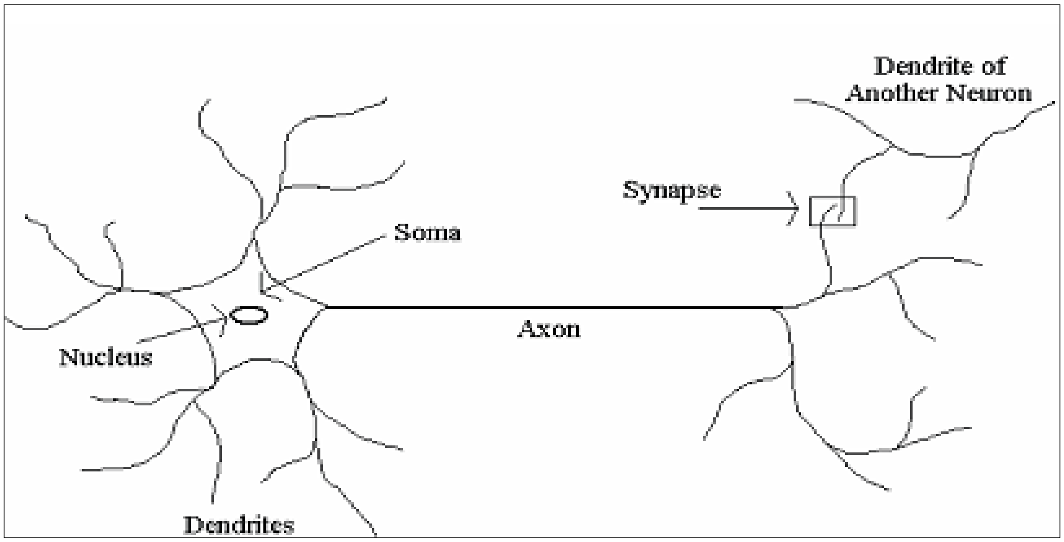


Figure2.1 Schematic diagram of a biological neuron

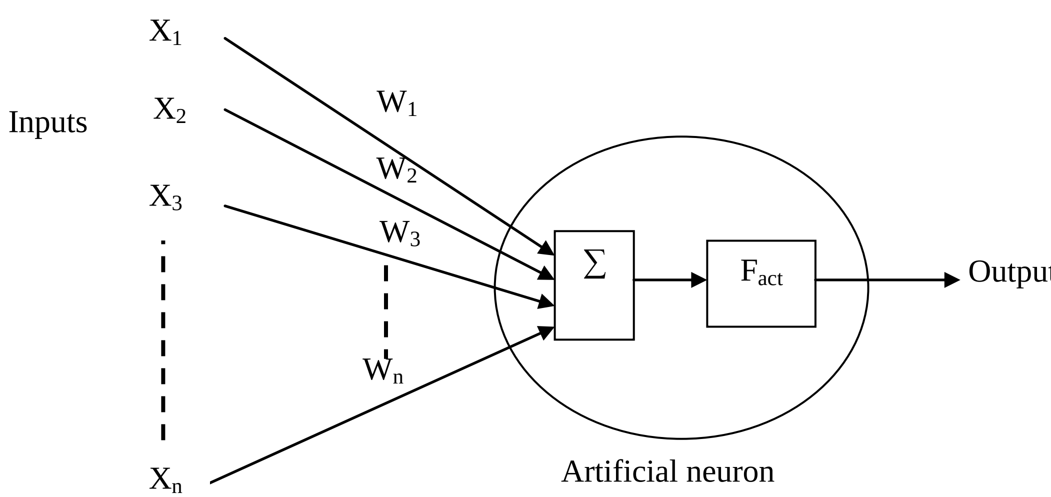


Figure 2.2 Schematic diagram of a simple artificial neuron

Where X_1, X_2 …X_n are the inputs, W_1, W_2, ….W_n are the weights, F_{act} is the activation function.

In general, the higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. The weights can also be negative, so the signal is inhibited by the negative weight. Computation of the neuron varies based on the weights. Outputs of artificial neuron for specific inputs can be obtained by adjusting the weights. When the neurons are less, it is easy to adjust the weights, but when size of neurons increases from hundreds to thousands, then it is quite complicated to find all the necessary weights by hand. However, in order to obtain the desired output from the network, researchers have explored several algorithms which will adjust the weights of ANN. This process of adjusting the weights is known as learning or training.ANNs gather knowledge by detecting the patterns and relationships in data and learn through experience.

The number of types of ANNs and their uses is increasing day by day. Different ANNs have different topology, the learning algorithms, etc. ANN with backpropagation algorithm (Rumelhart and McClelland, 1986) is widely used for learning the appropriate weights as it is one of the most common models used in ANNs, and many others are based on it. Since the function of ANNs is to process the information, they are used mainly in fields related with it. There are a wide variety of ANNs that are used to model real neural networks, such as behavior and control of machines. Also, there are ANN applications which are used in both Science (Medicine) and Technology (Engineering) such as pattern recognition, forecasting, data compression, signal processing, bio-technology and many more in multidisciplinary fields.

2.4 Selected ANN Applications for streamflow forecasting

A perceived strength of ANN is the capability for representing complex, nonlinear relationships as well as being able to model interaction effects. This capability is expected to be beneficial for forecasting since the relationship between the input variables and the resulting output (discharge) is typically quite complex. To explore the ability and capability of Artificial Neural Networks as an advance tool to use in the groundwater hydrology, in this direction a detailed literature survey has been carried out.

Christian W. Dawson et.al., (1998***),*** studied ANNs for flow forecasting in two flood-prone UK catchments using real hydrometric data. Given relatively brief calibration data sets it was possible to construct robust models of 15-min flows with six hour lead times for the Rivers Amber and Mole. Comparisons were made between the performance of the ANN and those of conventional flood forecasting systems. The results obtained for validation forecasts were of comparable quality to those obtained from operational systems for the River Amber. The ability of the ANN to cope with missing data and to "learn" from the event currently being forecast in real time is observed.

Thirumalaih and Deo (2000) found that ANN model performed better than statistical models like MR (Multiple Regression) and AR (Auto Regression) in hydrological forecasting.

F H. Keremet.al., (2003), low forecasting performance by artificial neural networks (ANNs) is generally considered to be dependent on the data length. In this study k-fold partitioning, a statistical method, was employed in the ANN training stage. The method was found useful in the case of using the conventional feed-forward back propagation algorithm. It was shown that with a data period much shorter than the whole training duration similar flow prediction performance could be obtained. Prediction performance and convergence velocity comparison between three different back propagation algorithms, Levenberg–Marquardt, conjugate gradient and gradient descent. The LM technique was found advantageous and more satisfactory performance criteria.

Deka and Chandramouli (2003) used neural network for deriving stage-discharge relationship in selected gauging sites of river Brahmaputra using twenty years daily observed time series data. They had developed modular neural network model considering the seasonwise flow data.They found that neural network model was better than other conventional models.

Anctil et al. (2004) examined the effect of data length using multiple-layer perceptions (MLP) and conceptual model. 1, 3, 5, 9, and 15year time sub-series created from a 24year training set, shifting by a 1-year sliding window to forecast 1-day aheadstream flow predictions. Based on their results, it is revealed that the MLP stream flow mapping wasefficient as long as wet weather data were available during training. Increases in the length of data cause the results to be consistent due to longer series of data which contains valuable information and gives clear information of hydrological behavior of a particular variable. However, it is plausible that a large number of internal parameters may allow better use of longer calibration series, but this was not verified in their study.

Nguyen and Chan (2004), carried out prediction study and found that data pre-processing is one of the most important steps for developing an ANN model for prediction.They have presented three data pre-processing strategies and gave the advantages, disadvantages and compare the results of each approach.

Kajitani et al. (2005) have studied the effect of different size of data sampling to the performance of ANN learning and generalization ability.

Jy S. Wu et al., (2005) Used ANN for watershed-runoff and stream-flow forecasts conducted on a small urban watershed in Greensboro, North Carolina. Two ANN-hydrologic forecasting models for watershed runoff prediction model to predict storm water runoff at a gauged location near the watershed outlet and another stream flow forecasting model was formulated to forecast river flows at downstream. Results obtained from both model applications are very encouraging even with a relatively small number of storm events employed for training and testing.

Deka and Chandramouli (2005)developed hybrid fuzzy neural network model for river flow prediction at downstream station of the Brahmaputra main stem river using upstream time series flow data as inputs for various combinations.They found that lagged upstream flow data influenced the model predictive performance for both ANN and hybrid model.

Ozgur Kişi et al., (2007), studied using ANN’s algorithms for short term daily streamflow forecasting. Four different ANN algorithms, namely, back propagation, conjugate gradient, cascade correlation, and Levenberg–Marquardt are applied to continuous streamflow data of the North Platte River in the United States. The models are verified with untrained data. The results from the different algorithms are compared with each other. The correlation analysis was used in the study and found to be useful for determining appropriate input vectors to the ANNs.

Ozgur Kişi., (2007), investigated the abilities of range-dependent neural networks (RDNN) to improve the accuracy of streamflow-suspended sediment rating curve in daily suspended sediment estimation. A comparison is made between the estimates provided by theRDNN and those of the following models: Artificial neural networks (ANN), linear regression (LR), range dependent linear regression (RDLR), sediment rating curve (SRC) and range-dependent sediment rating curve (RDSRC). The daily streamflow and suspended sediment data belonging to two stations-Calleguas Station and Santa Clara Station operated by the US Geological Survey were used as case studies. Based on comparison of the results, it is found that the RDNN model gives better estimates than the other techniques. RDLR technique is also found to perform better than the single ANN model.

Surinder Deswal et al., (2008*)*, studied an ANN based modeling technique to study the influence of different combinations of meteorological parameters on evaporation from a reservoir. Several input combination were tried so as to find out the importance of different input parameters in predicting the evaporation. The prediction accuracy of Artificial Neural Network has also been compared with the accuracy of linear regression for predicting evaporation. The comparison demonstrated superior performance of ANN over linear regression approach. The highest correlation coefficient (0.960) along with lowest root mean square error (0.865) was obtained with the input combination of air temperature, wind speed, sunshine hours and mean relative humidity. The findings of this study suggest the usefulness of ANN technique in predicting the evaporation losses from reservoirs.

Karim Solaimani (2009) utilized ANN for modeling the rainfall runoff relationship in a catchment area located in a semiarid region of Iran by adopting feed forward back propagation for the rainfall forecasting with various algorithms with performance of multi-layer perceptrons. The monthly stream of Jarahi Watershed was analyzed in order to calibrate of the given models. The monthly hydrometric and climatic were ranged from 1969 to 2000. The results extracted from the comparative study indicated that the ANN is more appropriate and efficient to predict the river runoff than classical regression model.

Mehdi Rezaeian Zadeh et al (2009), studied (ANN) models for predicting daily flows from Khosrow Shirin watershed located in the northwest part of Fars province in Iran. A Multi-Layer Perceptron (MLP) neural network was developed using five input vector using 5-year data record adopting Levenberg–Marquardt (LM) algorithm. It was found that antecedent precipitation and discharge with 1 day time lag as an input vector best predicted daily flows. Also, comparison of MLPs showed that an increase in input data was not always useful. The predicted outflow showed that the tangent sigmoid activation function performed better than did the logistic sigmoid activation function.

Mehmet C. Demirel et al., (2009), carried out study on the issue of flow forecast based on the soil and water assessment tool (SWAT) and artificial neural network (ANN) models. In this study, the ANNs were applied to the daily flow of the Pracana basin in Portugal. The comparison of ANN models and a process- based model SWAT was established based on their prediction accuracy. The ANN model was found to be more successful than the SWAT in relation to better forecast of peak flow. The SWAT model results revealed a better value of mean squared error. The study revealed that ANNs can be powerful tools in daily flow forecasts.

Lance E. Besaw et al., (2010) studied two ANNs to forecast streamflow in ungauged basins. The model inputs include time-lagged records of precipitation and temperature. In addition, recurrent feedback loops allow the ANN streamflow estimates to be used as model inputs. Streamflow records from sub-basins in Northern Vermontare used to train and test the methods. To predict streamflow in an ungauged basin, the recurrent ANNs are trained on climate-flow data from one basin and used to forecast streamflow in a nearby basin with different (more representative) climate inputs. One of the key results of this work is these recurrent flow predictions are being driven by time-lagged locallymeasured climate data. A scaling ratio, based on a relationship between bank full discharge and basin drainage area, accounts for the change in drainage area from one basin to another. Hourly streamflow predictions were superior to those using daily data for the small streams tested due the loss of critical lag times through up scaling. The ANNs selected in this work always converge, avoid stochastic training algorithms, and are applicable in small ungauged basins.

In spite of great efforts by the researchers in both traditional and soft computing techniques for time series forecasting, the need of producing higher accurate time series forecasts has motivated the researchers to develop new approach to model the time series. The current study aims to examine potential and applicability of ANN models in this situation for predicting flow using time series data. Also, we investigate the effect of data preprocessing on model performance in flow forecasting. Further, we explore the applicability of this network for river flow forecasting in different activation function for single step lead-time of one day ahead. Finally, we investigate the suitability of network for the site specific river flow prediction with acceptable and improved accuracy.

2.5 Outcome of Literature Review

Based on the literature review on ANN applications in hydrology it is observed that some of the grey area appears as mentioned below

- Few research works were carried out on streamflow forecasting using time series data. So, there is wide scope to develop different ANN networks using time series data for flow forecasting .

- A major concern for several researchers experienced in different application of ANN is the lack of quality and quantity of the required data, detailed information of the system or problem and data size of effective domain in time series.
- Using streamflow as the sole variable for prediction is undertaken by very few researchers.

- Log transform and Log plus First Difference are untried new preprocessing techniques for time series.

To address above limitations, an attempt has been made to improve the forecasting accuracy of streamflow using various data preprocessing techniques in various Artificial Neural Network architectures for various input scenarios with time series flow data at temporal scale. Also, stress is given for forecasting accuracy as it is one of the important factors involved in selecting a forecasting method.

CHAPTER - 3

MATERIALS AND METHODOLOGY

3.1 Introduction

While developing ANN models of the hydrologic time series, most of the researchers have employed raw data to be presented to the ANN. The raw data consist of various trends in the form of long term memory and seasonal variations. For these reasons, the hydrologic time series may be nonstationary affecting the performance of the ANN models. It may be possible to improve the performance of ANN models by first carefully removing the long term and seasonal variations before presenting an ANN with the modified data.

In order to improve the performance of any model, the model requires sufficient amount of input data. In this type of situation, it is often difficult to obtain reliable forecasts of future river flow, due to the lack of accurate data for the required model inputs. The remote location and complex hydraulic relationships of many of the sites contribute to a poor quality of river flow monitoring. The advance tool such as ANN has been found to be effective and more efficient in situations where noisy data attached with shorter length of observed data.

3.2Classification and selection of data

At Pandu station, the data points are arranged for one day, two day and three day lagged data, which gives 6936 data points. These are divided in the ratio of beginning 2/3rd for ‘Training and Validation Dataset’ and remaining 1/3rd for ‘Testing Dataset’. Thus we have used 4624 data points for training and validation and 2312 data points for testing.Similarly for Pancharatna station, 7302 data points are available out of which

4868 data points used for training and validation and 2434 data points for testing.The statistical characteristics of flow data are shown in the tables below which reveals high variability of the flow data.

Table 3.1 Statistical Charachteristics of Data at Pandu

Station Pandu	**Statistical Parameters**	**Training Set Q (m^3/s)**	**Testing Set Q (m^3/s)**	**All Data Set Q (m^3/s)**
DAILY DATA	Min	2432	5539	2432
	Max	61015	54100	61015
	Mean	17520	18897	17904
	Standard Deviation S_d	11011	10306	10836
	Skewness Coefficient C_{xx}	0.59	0.64	0.59

Table 3.2 Statistical Characteristics of Data at Pancharatna

Station Pancharatna	**Statistical Parameters**	**Training Set Q (m^3/s)**	**Testing Set Q (m^3/s)**	**All Data Set Q (m^3/s)**
DAILY DATA	Min	2086	1723	1723
	Max	75277	76236	76236
	Mean	16486	16550	16503
	Standard Deviation S_d	13771	12086	13192
	Skewness Coefficient C_{xx}	0.99	0.81	0.95

Data at both the stations show a high skewness coefficient. It is observed that high skewness coefficient has a considerable negative effect on ANN performance (Altun et al, 2007). The standard deviation is also large and the data range is also large approximately between 10^3 and $8x10^5$. Thus the characteristics of the data merit the consideration of using pre-processing techniques before presenting them to the ANNs as input.

3.3 Visual Observation of Data

The data at both the stations are plotted for visually observing and getting an insight into the nature of the data, trends and seasonality. Following pages represent the plots of the data at the two stations.

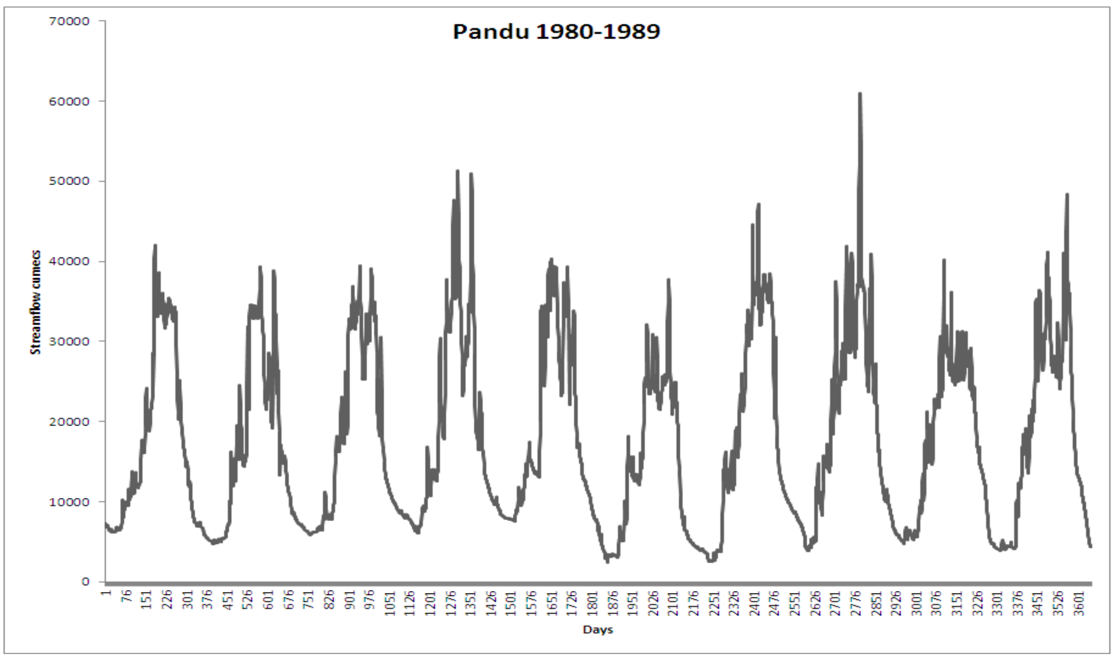


Fig.3.1 Daily Streamflow at Pandu 1980-1989

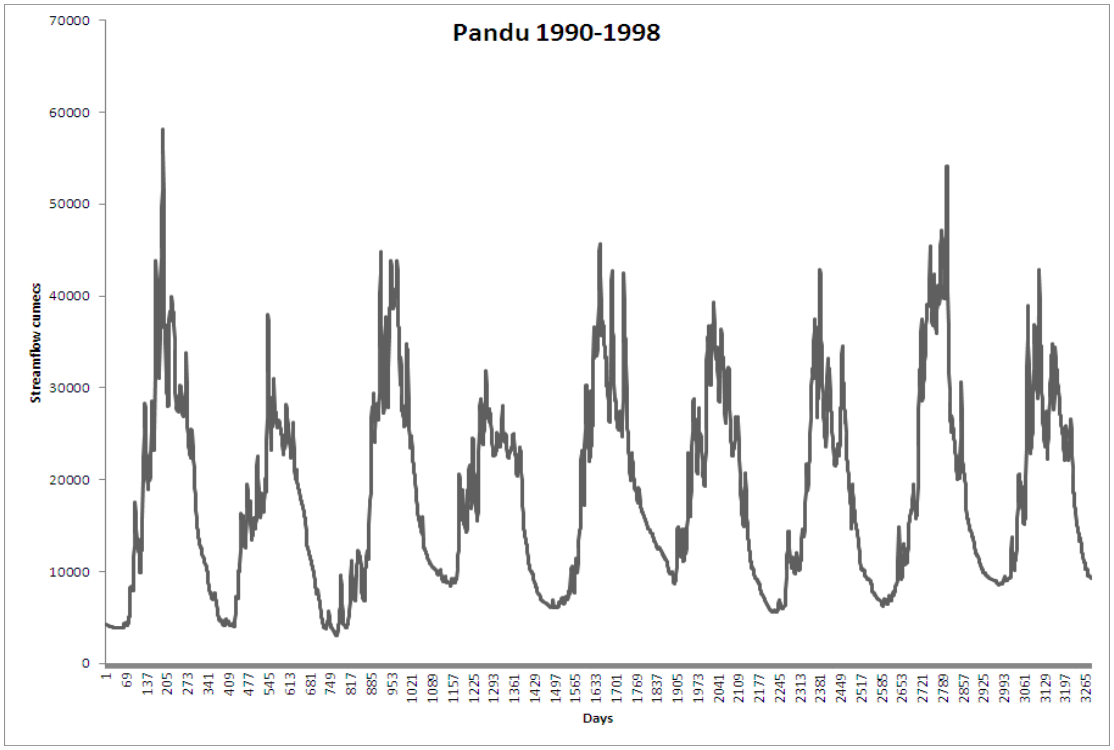


Fig. 3.2 Daily Streamflow at Pandu 1989-1998

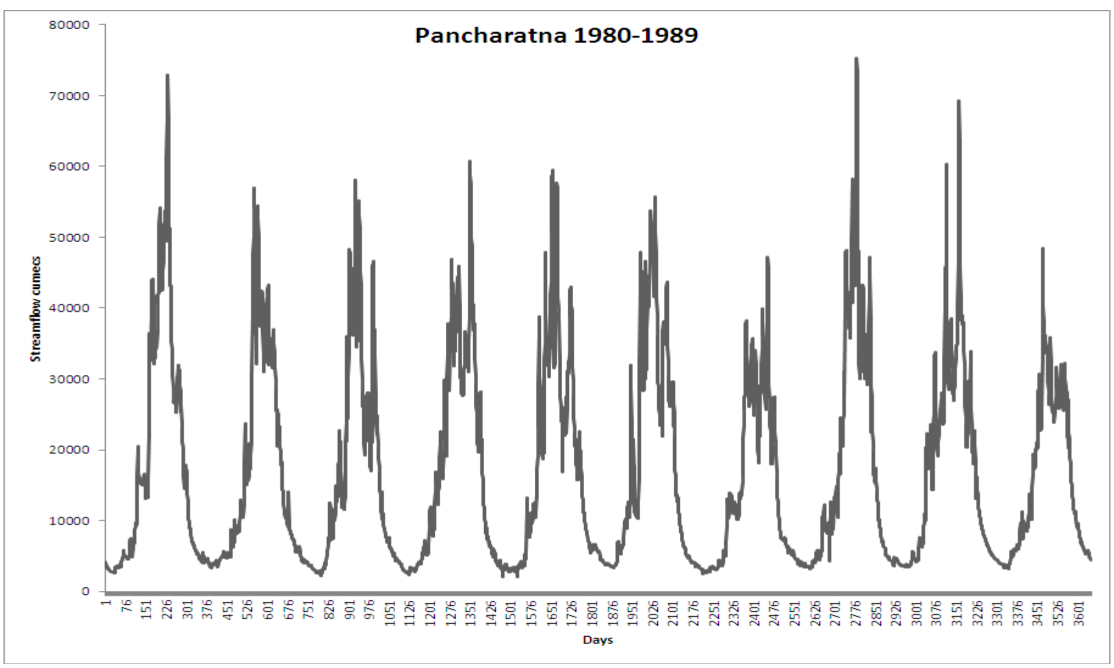


Fig. 3.3 Daily Streamflow at Pancharatna 1980-1989

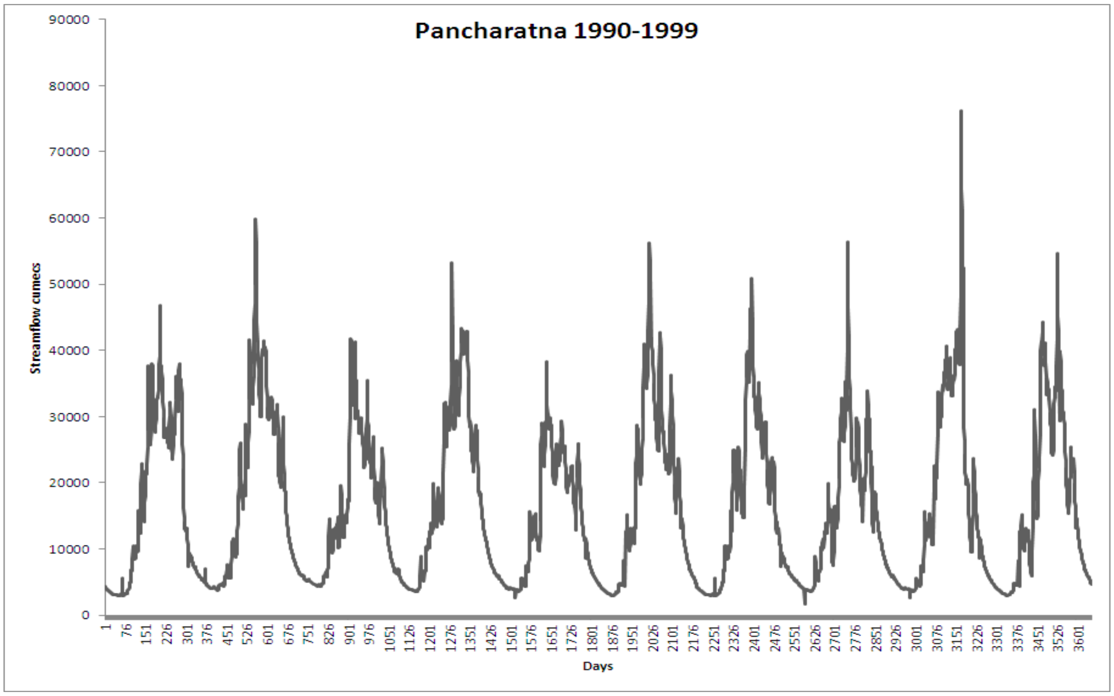


Fig. 3.4 Daily Streamflow at Pancharatna 1990-1999

Yearly flooding pattern during the monsoon months of May to July is seen at both the stations.

3.4 Artificial Neural Networks

Artificial Neural Networks are mathematical inventions inspired by observations made in the biological systems.ANN has gained popularity among Hydrologist in recent decades due to its large array of application in the field of Engineering and research. The first neuron was produced in 1943 by the neurophysiologist Warren McCullo and the logician Walter Pits. Thereafter, till 1969 Minsky and Papert wrote a book in which they generalized the limitations of Artificial Neural Networks. The era of renaissance started with John Hopfield in 1984 introducing recurrent neural network architecture.

The purpose of ANN is mapping function i.e., mapping an input space to an output-space.ANN has excellent flexibility and high efficiency in dealing with nonlinear and noisy data in Hydrological modeling. Some of the advantages of using ANN Tool are Input-Output mapping, Self-adaptive, Real-Time Operation, Fault Tolerance and Pattern Recognition. A typical ANN consists of a number of nodes that are organized according to a particular arrangement. It consists of “Neurons” which are interconnected computational elements that are arranged in a number of layers which can be single or multiple.

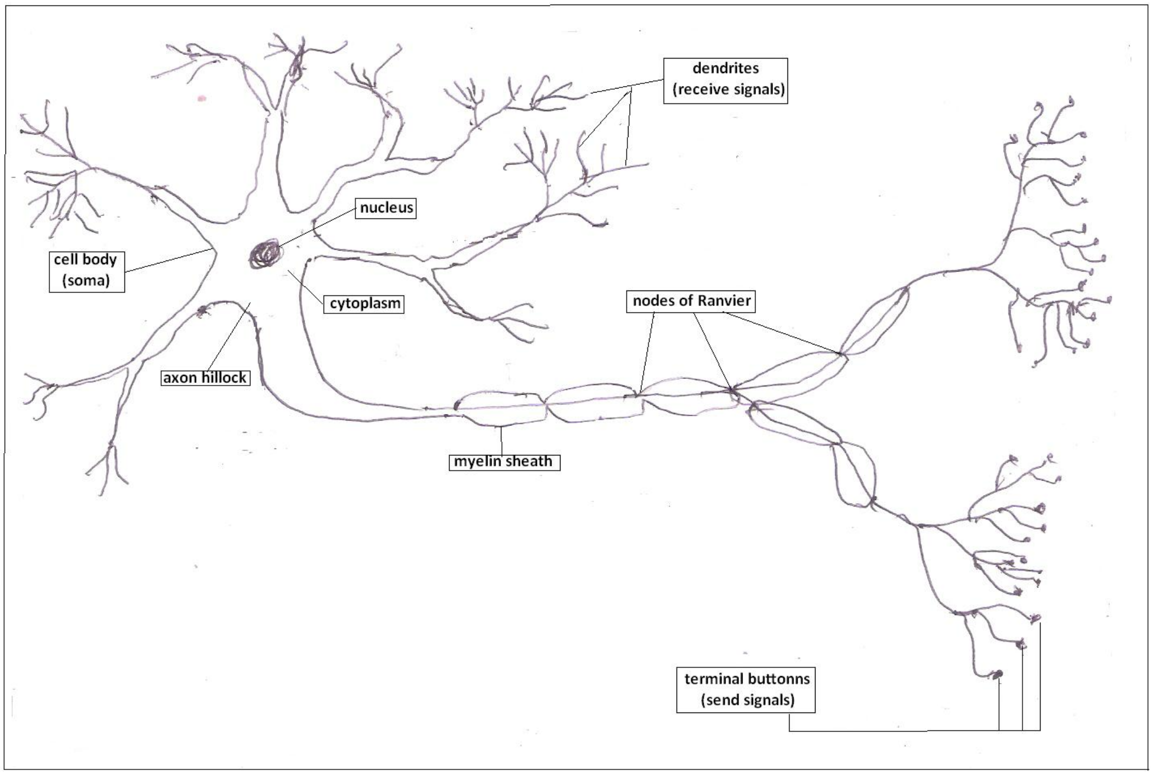


Fig 3.5 A Typical structure of a neuron

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.

The human nervous system may be viewed as a three-stage system receptors, neural net and effectors whenever a stimulus is generated followed by response which is the output. Central to the system is the brain, represented by the neural (nerve) net, which continually receives information, perceives it, and makes appropriate decisions

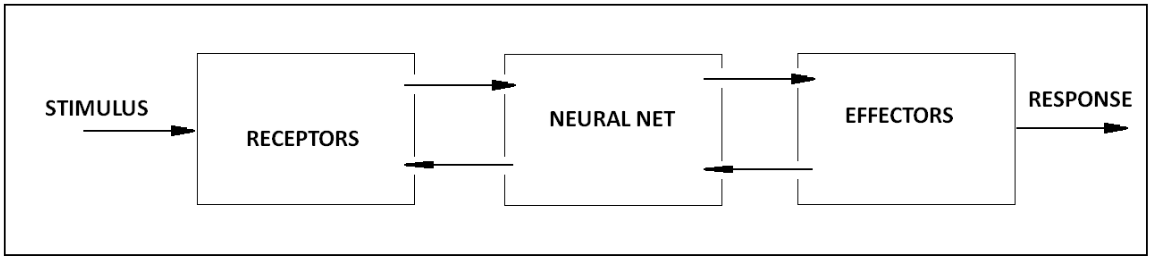


Fig 3.6 Block diagram representation of nervous system

Two sets of arrows are shown in the block diagram of nervous system. Those pointing from left to right indicate the forward transmission of information- bearing signals through the system. The arrows pointing from right to left signify the presence of feed-back in the system. The receptors convert stimuli from the human body or the external environment into electrical impulses that covey information to the neural net (brain). The effectors convert electrical impulses generated by the neural net into discernible responses as system outputs.

A typical ANN consists of a number of nodes that are organized according to a particular arrangement. It consists of "Neurons" which are interconnected computational elements that are arranged in a number of layers which can be single or multiple. A neuron is an information-processing unit that is fundamental to the operation of a neural network. Each pair of neurons is linked and is associated with weights.

The three basic elements of the neuronal model are

1) **Synapses**: A set of synapses or connecting links, each of which is characterized by the weight or strength of its own. Specifically, a signal $x_j$at the input of synapse*j*connected to neuron *k* is multiplied by synaptic weight $w_{kj.}$ Here the subscript 'k' refers to the neurons in question and the subscript 'j' refers to the input end of the synapse to which the weight refers. Unlike a synapse in the brain, the synaptic weight of an artificial neuron may lie in a range that includes negative as well as positive values.

The block diagram figure 3.6 shows the model of a neuron which forms the basis for designing (artificial) neural networks.

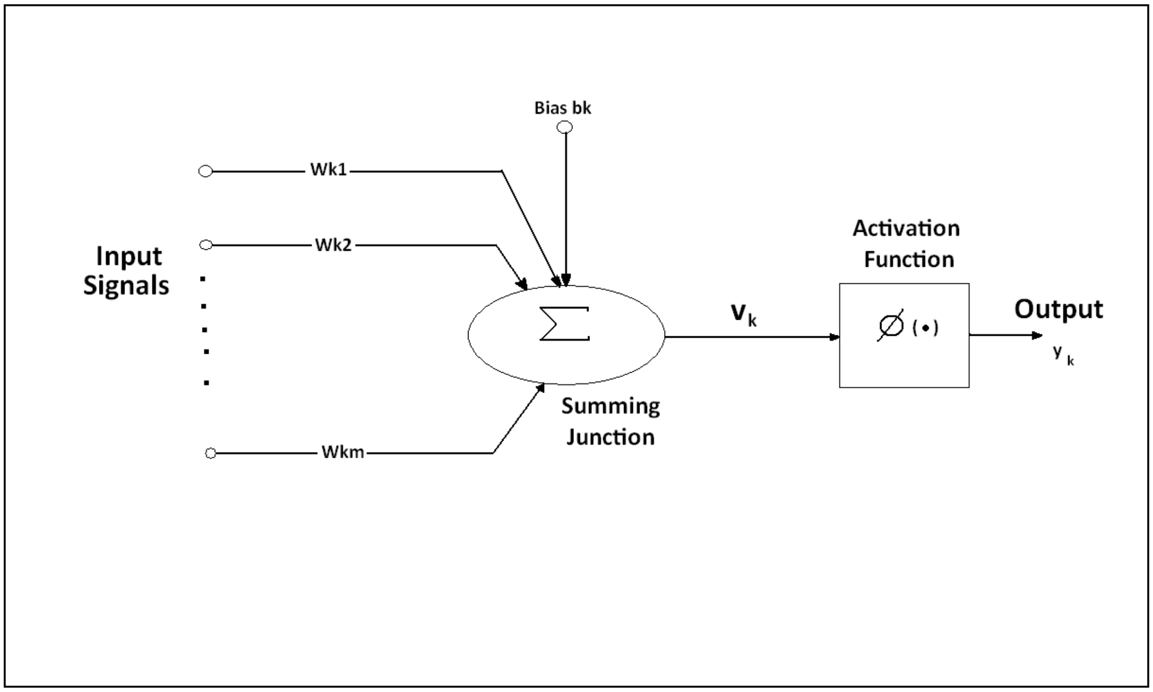


Fig 3.7 Nonlinear model of a neuron

Courtesy :SymondHykins

2) **An adder** for summing weight of an artificial neuron may lie in a range that includes negative as well as positive values.
3) **An Activation Function** for limiting the amplitude of the output of a neuron. The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1]. The neuronal model as shown in fig 3.3 also includes an externally applied ***bias*** denoted by b_k. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively.

In mathematical terms, a neuron '*k*' can be written in the form of the equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

and ……………….. (3.1)

$$y_k = \varphi\ (u_k + b_k) \qquad \ldots\ldots\ldots\ldots\ldots\ldots..\ (3.2)$$

Where x_1, x_2,........x_m are the input signals; w_{k1}, w_{k2},...., w_{km} are the synaptic weights of neuron k; $u_k$is the linear combiner output due to the input signals; b_k is the bias; φ(•) is the activation function; and 'y_k' is the output signal of the neuron. The use of bias $b_k$has the effect of applying an affine transformation of the output $u_k$of the linear combiner in the model of fig 3.5 and is show by

$$V_k = u_k + b_k \qquad \ldots\ldots\ldots\ldots\ (3.3)$$

Depending up whether the bias $b_k$is possible or negative, the relationship between the induced local field or activation potential v_k of neuron k and linear combiner output u_k is modified.

3.5 Types of Activation Function

The behavior of an ANN depends on both the weights and the input-output function (transfer function) that is specified for the units

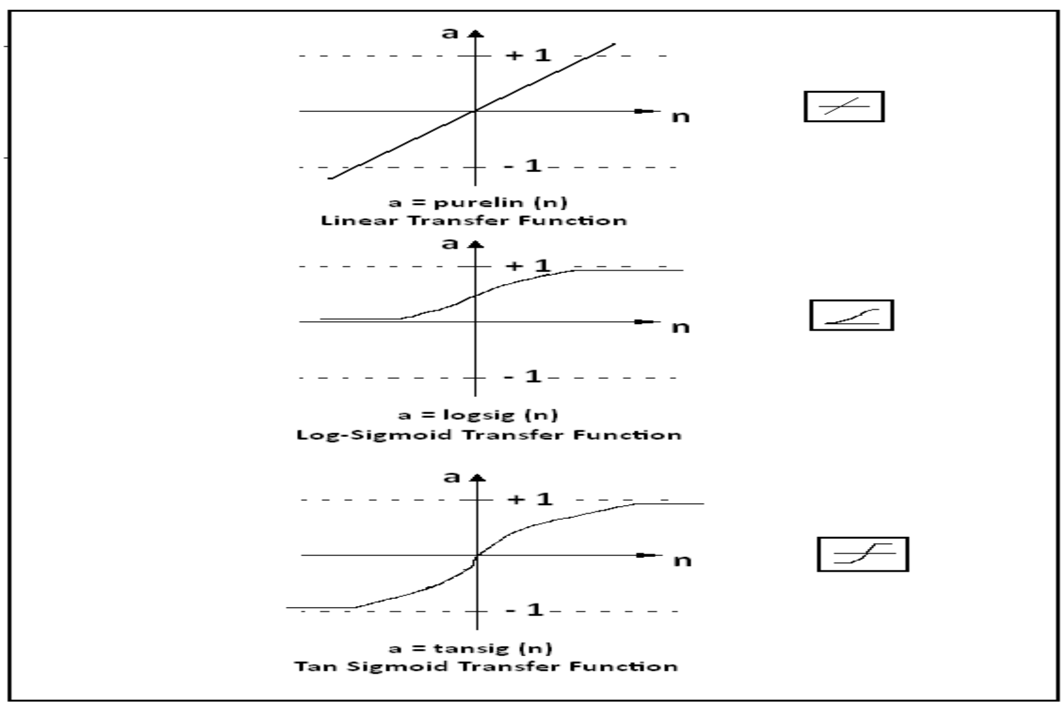


Fig 3.8 Types of activation function

This function typically falls into one of three categories:

1. **Linear (or ramp):** The output activity is proportional to the total weighted output.
2. **Threshold:** The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.
3. **Sigmoid:** The output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

3.6ANN Architecture

ANN Architecture consists of three-layer i.e., the Input-Layer, the Hidden-Layer and the Output-Layer. The network consists of three distinctive modes: training, cross-validation and testing. The behavior of an ANN depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories- Linear (or ramp), Threshold and Sigmoid**.** An important step in developing an ANN model is the determination of its weight matrix through training. There are primarily two types of training mechanisms, supervised and unsupervised.

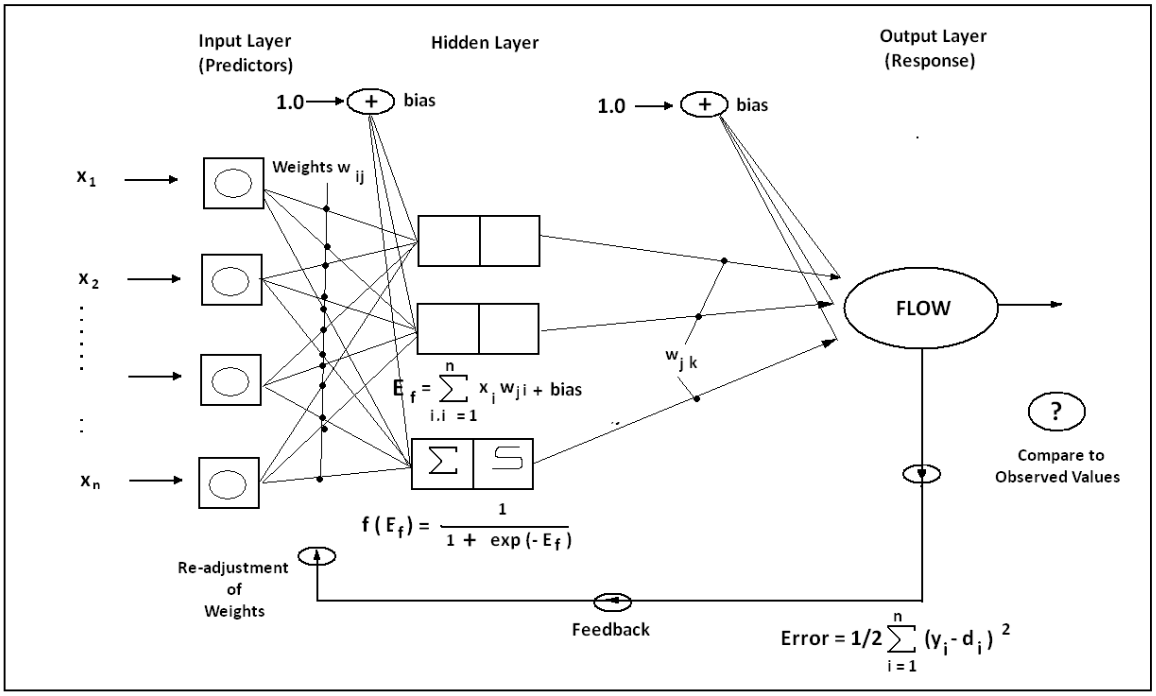


Fig 3.9 Three layered FFNN with BP training algorithm

a) Supervised Training

A supervised training algorithm requires an external teacher to guide the training process. The primary goal in supervised training is to minimize the error at the output layer by searching for a set of connection strengths that cause the ANN to produce outputs that are equal to or closer to the targets. A supervised training mechanism called back-propagation training algorithm is normally adopted in most of the engineering applications.ANN is trained by adjusting the values of these connection weights between network elements. The weighted inputs in each layer are processed from neurons in the previous layer and transmit its output to neurons in the next layer. A transfer function is used to convert a weighted function of input to get the output.

b) Unsupervised Training

Another class of ANN models that employ an 'unsupervised training method' is called a self-organizing neural network. The data passing through the connections from one neuron to another are multiplied by weights that control the strength of a passing signal. When these weights are modified, the data transferred through the network changes; consequently, the network output also changes. The signal emanating from the output node(s) is the network's solution to the input problem. Each neuron multiplies every input by its interconnection weight, sums the product, and then passes the sum through a transfer function to produce its result. This transfer function is usually a steadily increasing S-shaped curve, called a sigmoid function.

3.6.1 Learning Process

Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. This definition of the learning process implies the following sequence of events:

1. The neural network is stimulated by an environment.
2. The neural network undergoes changes as a result of this stimulation.
3. The neural network responds in a new way to the environment, because of the changes that have occurred in its internal structure.

Let $w_{kj}(n)$ denote the value of the synaptic weight w_{kj}at time n. At time n an adjustment $\Delta w_{kj}(n)$ is applied to the synaptic weight w_{kj} (n), yielding the updated value

$$\mathbf{W_{KJ}(N+1) = W_{KJ}(N) + \Delta W_{KJ}(N)} \qquad \textbf{………………(3.4)}$$

A prescribed set of well-defined rules for the solution of a learning problem is called a learning algorithm. As one would expect, there is no unique learning algorithm for the design of neural networks. Rather, we have a “kit of tools” represented by a diverse variety of learning algorithms, each of which offers advantages of its own. Basically, learning algorithms differ from each other in the way in which the adjustment Δw_{kj} to the synaptic weight w_{kj} is formulated.

3.6.2 Feed forward Back propagation

Multilayer perceptions have been applied successfully to solve some difficult diverse problems by training them in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm. This algorithm is based on the error-correction learning rule. Basically, the error back-propagation process consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network, layer by layer. Finally, a set of outputs is produced as the actual response of the network.

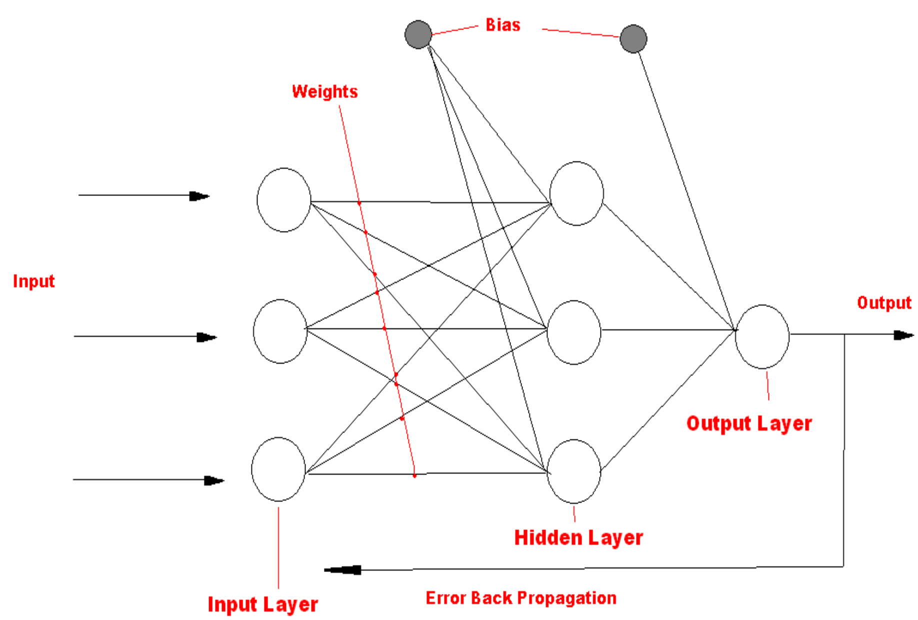


Fig 3.10 Back propagation

During the forward pass the synaptic weights of network are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with the error-correction rule. Specifically, the actual response of the network is subtracted from a desired (target) response to produce an error signal. This error signal is then propagated backward through the network, against direction of synaptic connections

- hence the name “error back-propagation”. The synaptic weights are adjusted so as to make the actual response of the network move closer the desired response. The error back propagation algorithm is also referred to in literature as the back-propagation algorithm, or simply back-prop. The feed-forward back-propagation neural network in Figure 3.10 is fully connected, which means that a neuron in any layer is connected to all neurons in the previous layer. Signal flow through the network progresses in a forward direction, from left to right and on a layer-by layer basis. The connection weights manifest the importance of input to the overall estimation process. The fitting error Eq. (3.5)between thedesired and estimated output is used as feedback to enhance the performance of the network by altering the connection weights:

$$Error = \sum_{j=1}^{N} (y_j - d_j)^2 \qquad (3.5)$$

Where, N = number of output nodes, y_j= calculated output, and d_j = desired data value.

This process is repeated until establishing a successive layer. Therefore, these kinds of networks are called feed forward back propagation (FF-BP) networks, which are the most popular supervised algorithm for training networks in prediction, pattern recognition, and nonlinear function fitting. Training (calibrating) is a crucial process, in which the network is tested by a set of data pairs (input–output)and changing the initial conditions in each iteration step to achieve an accurate forecasting. Minimization is performed by calculating the gradient for each node at the output layer.

$$\delta_k = d\sigma_k(y_k - d_k) \qquad (3.6)$$

d_{rk} = the derivative of the sigmoid function applied at y_k which is defined for each k^{th} output node. For hidden layer (one layer back),the gradient function becomes

$$\delta_j = d\sigma_j \sum_{i=1}^{N} \delta_i W_{jk} \qquad (3.7)$$

Where d_{rj} is the derivative of the sigmoid function and w_{jk} = weight value from hidden node j to output node k. When the input data are chosen, then the network runs; the weights for each connection are updated by the procedure in Eq. (3.6) until the error is minimized to a predefined error target or the desired number of training periods is reached:

$$\Delta W_{jk} = W_{jk} - \eta \delta_k y_j \tag{3.8}$$

Where, 'η' the notation g is the learning rate of each layer back to the network. Each passes through the training data is called epoch. In the Matlab routines, the user can define the number of epochs prior to analysis and manually adjusts until the plausible performance is achieved in the trial and error period.

The use of neural networks offers the following useful properties and capabilities:

1. **Nonlinearity:** A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear. Moreover, the nonlinearity is of a special kind in the sense that it is distributed throughout the network.

2. **Input-output mapping:** A popular paradigm of learning called supervised learning involves the modification of the synaptic weights of a neural network by applying a set of training samples. Each sample consists of a unique input signal and the corresponding desired response. The network is presented a sample picked at random from the set, and the synaptic weights (free parameters) of the network are modified so as to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate criterion. The training of the network is repeated for many samples in the set until the network reaches a steady state, where there are no further significant changes in the synaptic weights. The previously applied training samples may be re-applied during the training session, usually in a different order. Thus the network learns from the samples by constructing an input-output mapping for the problem at hand.

3. Adaptability: Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a non-stationary environment a neural network can be designed to change its synaptic weights in real time. The natural architecture of a neural network for pattern classification, signal processing, and control applications, coupled with the adaptive capability of the network, makes it an ideal tool for use in adaptive pattern classification, adaptive signal processing, and adaptive control.

4. Contextual information: Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally by a neural network.

5. Fault tolerance: A neural network, implemented in hardware form, has the potential to be inherently fault tolerant in the sense that its performance is degraded gracefully under adverse operating. For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality. However, owing to the distributed nature of information in the network, the damage has to be extensive before the overall response of the network is degraded seriously. Thus, in principle, a neural network exhibits a graceful degradation in performance rather than catastrophic failure.

6**. VLSI Implementability:** The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network ideally suited for implementation using very-large-scale-integrated (VLS1) technology.

7. Uniformity of analysis and design: Basically, neural networks enjoy universality as information processors. We say this in the sense that the same notation is used in all the domains involving the application of neural networks. This feature manifests itself in different ways:

a) Neurons, in one form or another, represent an ingredient common to all neural networks.

b) This commonality makes it possible to share theories and learning algorithms in different applications of neural networks.

c) Modular networks can be built through a seamless integration of modules.

8.Neurobiological analogy: The design of a neural network is motivated by analogy with the brain, which is a living proof that fault-tolerant parallel processing is not only physically possible but also fast and powerful. Neurobiologists look to (artificial) neural networks as a research tool for the interpretation of neurobiological phenomena. On the other hand, engineers look to neurobiology for new ideas to solve problems more complex than those based on conventional hard-wired design techniques. The neurobiological analogy is also useful in another important way: It provides a hope and belief that physical understanding of neurobiological structures could influence the art of electronics and thus VLSI.

3.6.3 Strengths of ANN

1. ANNs are better in terms of result accuracy than almost all prevalent analytical, statistical or stochastic schemes (Jain and Deo, 2004).
2. ANNs methodologies have been reported to have capability of adapting to a nonlinear and multivariate system having complex inter-relationships which may be poorly defined and not clearly understood using mathematical equations (Thirumalah and Deo, 1998).
3. Input data that are incomplete and ambiguous or data with noise, can be handled properly by ANNs because of their parallel processing (Flood and Kartam-I, 1993; ASCE, 2000a).
4. ANNs are able to recognize the relation between the input and output variables without explicit physical consideration of the system or knowing underlying principle because of the generalizing capabilities of the activation function (ASCE, 2000a; Thirumalah and Deo, 1998).
5. Accuracy of ANNs increases as more and more input data is made available to it (Tokar and Markus, 2000)
6. The time is consumed in arriving at best network and training but ANNs once trained, are easy to use. It is much faster than a physical based model which it approximates (ASCE, 2000a).
7. ANNs are able to adapt to solutions over time to compensate for changing circumstances (suitable for time variant problems).

8. ANNs are more suitable for dynamic forecasting problems because the weights can be updated when fresh observations are made available (Thirumalah and Deo, 1998).
9. Neural networks can be complimentary or alternative to many complex numerical schemes including FEM/FDM (Jain and Deo, 2004).

3.6.4 Weakness of ANN

1. ANN's extrapolation capabilities, beyond its calibration range, are not reliable. During prediction ANN is likely to perform poorly if it faces inputs that are far different from the examples it is exposed to during training. Therefore prior information of the system is of utmost importance to obtain reasonably accurate estimates (ASCE, 2000a).
2. It is not always possible to determine significance of the input variables prior to the exercise and it is important to identify and eliminate redundant input variables that do not make a significant contribution to the model. This would result in a more efficient model.
3. The knowledge contained in the trained networks is difficult to interpret because it is distributed across the connection weights in a complex manner.
4. The success of ANN application depends both on the quality and quantity of data available (ASCE, 2000a), type and structure of the neural network adopted and method of training (Flood and Kartam-II, 1993).
5. Determining the ANN architecture is problem dependent trial and error process (Shigdi and Gracia, 2003). The choice of network architecture, training algorithm and definition of error are usually determined by the users past experience and preference, rather than the physical aspects of the problem (ASCE, 2000a).
6. Initialisation of weights and threshold values are an important consideration (Kao, 1996). This problem is faced particularly while implementing back propagation training algorithm. Some of the researchers have tried to overcome this problem by using genetic algorithm (GA) global search method.

7. While training the network there is a danger of reaching local optimum especially for backpropagation algorithm. Global search techniques like genetic algorithm and simulated annealing are useful in such conditions.
8. Representing temporal variations is often achieved by including past inputs/outputs as current inputs. However it is not immediately clear how far back one must go in the past to include temporal effects.

3.7 **Overview of Research Methodology Adopted**

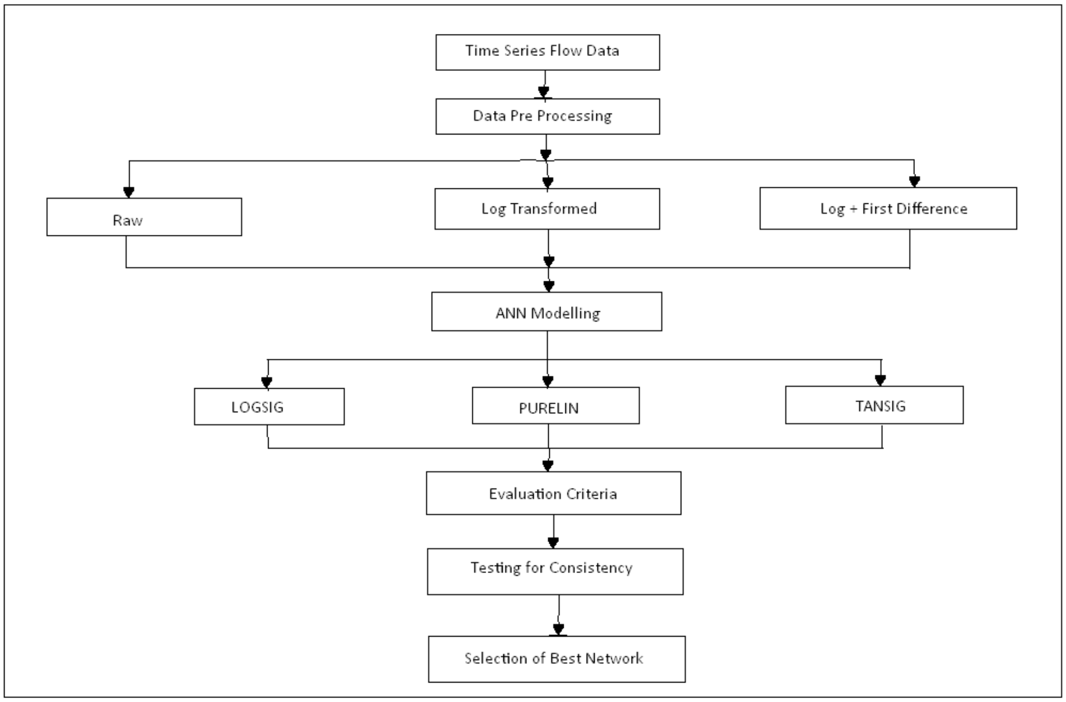


Fig. 3.11 Flow Chart of Methodology

Three architectures of the ANN of the type Multi-Layer Perceptron (MLP)areused.The Feed Forward Error Back Propagation (FFBP) algorithm is used. The MLP has three layers, viz, Input layer, Hidden layer and Output layer. The neurons of the hidden layer are varied from 1 to 10. The architectures PURELIN, LOGSIG and TANSIG are used. These are according to the activating function of the ANN.

The equations for transformation of the three functions are given below:

Purelin function $y = x$

Logsig function $y = 1/(1 + e^{-x})$

Tansig function $y = \tanh (e^{x} - e^{-x})/(e^{x} + e^{-x})$

Where x is the total of all weighted inputs alongwith bias at a node and y is the output from that node.

3.8 Training of ANN

During the training of ANN the input is given to the network as well as the target is provided. These are from the 1/3rd of total data points forming the training dataset. The prediction by the network gets compared to the target and the error is back-prpagated. This is done through the Graphical User Interface (GUI) by importing the appropriate values from the workplace to the NN tool specifying the category and then constructing a network from the GUI with required architecture. The number of neurons for the network as also the number of layers in the MLP, all this can be changed according to requirement from the GUI of the NN tool in the software Matlab.

3.8.1 Validation

For validation, the same data set as used for training is given as the input, only this time, the targets are not provided and the now trained network predicts the outputs according to the weight matrix formed during the training process. Here the errors are not calculated and no backpropagation occurs. Back propagation occurs only during the training process.

3.8.2Testing

In the testing process, the remaining 2/3rd datapoints forming the testing datasets are used. An already trained network is given the testing data as input and it predicts the output. The data are totally unseen by the network and the prediction is done through the weight matrix of the interconnections acquired during the training process.

3.8.3Feed-forward back propagation-Levenberg-Marquardt (FFBP-LM)

Here, the Feed Forward Back Propagation (FFBP) neural network was trained using Levenberg-Marquardt (LM) technique because it is more powerful and faster than the conventional gradient descent technique (Hagan and Menhaj; 1994; Kisi, 2007). The LM algorithm was designed to approach second order training speed without having to compute the Hessian matrix (More, 1977). The Levenberg-Marquardt method is a standard technique used to solve nonlinear least squares problems. Nonlinear least squares problems arise when the function is not linear in the parameters. Nonlinear least squares methods involve an iterative improvement to parameter values in order to reduce the sum of the squares of the errors between the function and the measured data points. It has become a standard technique for nonlinear least-squares problems and can be thought of as a combination of two minimization methods steepest gradient descent and the Gauss-Newton method. The Levenberg-Marquardt curve-fitting method is actually a combination of two minimization methods: the gradient descent method and the Gauss-Newton method. The performance function will always be reduced at each iteration of the algorithm. The application of LM to neural network training is described in Hagan and Menhaj (1994). Schematic diagram of feedforward neural network are shown in Figure 3.12.

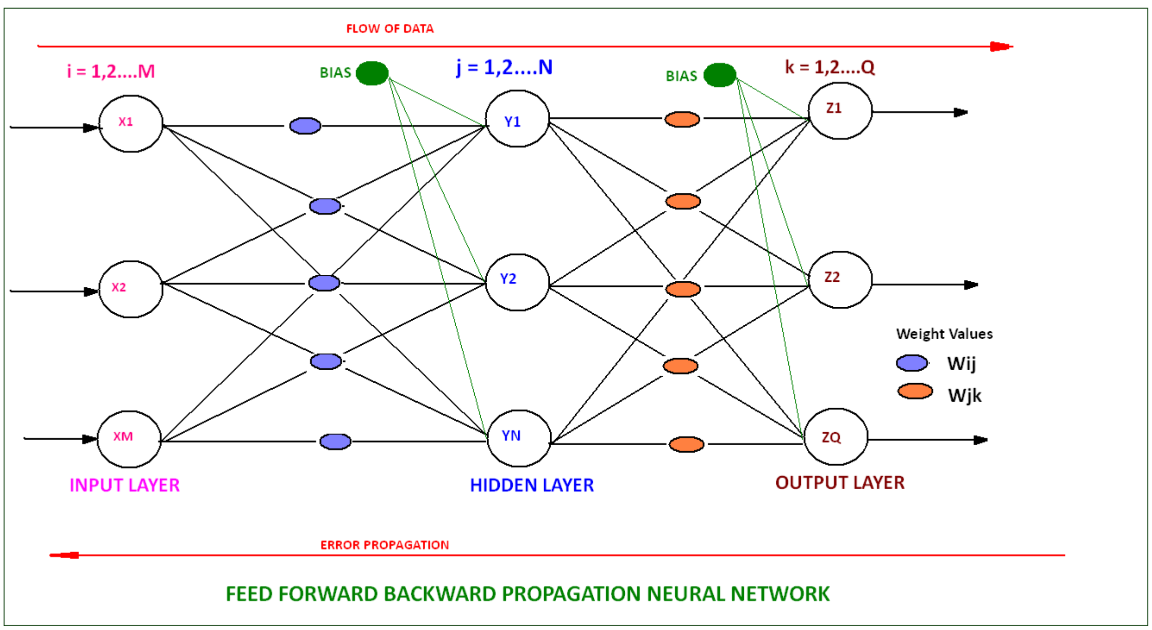


Figure 3.12 Schematic diagram of feedforward neural network

The Feed forward back propagation algorithm (FFBP) has been widely used in hydrology because of its simplicity, robustness and the advent of error back propagating. Basically this algorithm consists of two phases. In the forward pass the input signals propagate from the network input to the output. In the backward or reverse pass, the calculated error signals propagate backward through the network, where these are used to adjust the weights. Because of this reason several researchers have tried with back propagation for forecasting purposes. FFBP can be found in detail manner (Haykin 1999).

3.9 Total Number of Trials

For each architecture, number of neurons are varied from 1 to 10. This accounts for 30 types of networks. There are three types of datasets, Raw, Log transformed and Log plus first difference. In eac dataset, number of lagged terms varies from 1 to 3. Thus there are 9 inputs for each type of datasets, making a total of

30 network types X 9 input datasets = 270 trials.

Thus ther are 2 X 270 trials as each trial is conducted for validation as well as for testing. Thus at each station there are 540 trials conducted.

This makes the total number of trials at Pandu and Pancharatna as 1080 trials.

3.10 Software Used

Matlab R2010a was used for the network analysis and MS Excel, MS Word, MS Paint were used for compilation and presentation.The figures below show the screenshot of theGUI of the Matlab workspace and the NN tool.

3.11 Evaluation Criteria

The selection of best network is done primarily on testing results as these are unseen by the trained network and hence closer to the actual field situation. When testing and validation results of a network show very large discrepancy, that network is discarded and the next best is selected.

Studying the protocol for network evaluation as given by Dawson and Wilby (2001), it is decided to use a combination of RMSE and MAPE, as the evaluation criteria as suggested by Legates and McCabe (1999).

RMSE means the Root Mean Squared Error and is given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(X - Y\right)^2}{N}}$$

Where X and Y are observed and computed values respectively and N is the total number of observations. Thus the drawback of cancellation of positive and negative errors with each other and returning a high coefficient of correlation R^2 leading to erroneous assessment is avoided here. The other favorable feature of RMSE is that it gives the error in the same units as the input and output hence a realistic assesment becomes easier.

MAPE means the Mean Absolute Percentage Error and is given by

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|X - Y|}{X}\times 100$$

Thus MAPE gives the proportionate nature of error in relation with the input data and hence it is easier to correlate with performance especially when the range of data is large as in this work.

3.12 Conclusion

The entire process is described for the sake of clarity and so that anyone can entirely build up and check upon the work presented here starting from the original data files. Also the details of the configuration used are in keeping with the protocol for ANN research suggested by Dawson and Wilby (2001).

The materials, their classification and the methodology of using the Artificial Neural Networks is explained giving some basic idea of the processes involved in ANN technique.

CHAPTER – 4

RESULTS AND DISCUSSIONS

4.1. Introduction

The results of the various network trials conducted with different combinations of raw and pre-processed data sets with different network architectures and varying number of neurons are presented here.

4.2 Results for Pandu Station

These are enumerated for the three types of datasets, viz. Raw, Log Transformed and Log plus First Difference.

4.2.1 Raw Data Sets - One Day Lag

Here the raw data of streamflow is given to the network as input and the network predicts the streamflow of the next day. The following table shows the evaluation results of the 30 trials resulting from the three architectures and by varying neurons in the hidden layer from 1 to 10.

After reaching the minimum value of 4.87 for MAPE in the testing dataset and 3.46 in the training data sets, any further addition of neurons does not reduce the error and in fact in some trials increases it. This is typical feature observed in ANNs that up to a certain limit the addition of neurons improves the performance but further increase may cause problems associated with overtraining and overfitting where the ANN tries to predict each individual value correctly and loosing the track of the pattern that the data follows, fails to improve the performance.

Table 4.1 Raw Data 1Day Lag –LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1495.20	9.51	1834.72	6.15
2	17100.53	70.82	17645.51	77.73
3	1185.89	3.49	1725.93	4.93
4	1199.29	3.46	1765.99	4.87
5	1186.97	3.62	1731.27	4.94
6	1198.90	3.49	1759.52	4.91
7	1184.17	3.53	1728.81	4.95
8	1184.92	3.48	1735.99	4.89
9	1188.39	3.58	1744.73	4.91
10	1179.85	3.49	1743.16	4.89

The abbreviations in this and following tables are as follows:

RMSE TR : Root Mean Squared Error for Training and Validation Dataset

RMSE TST : Root Mean Squared Error for Testing Dataset

MAPE TR : Mean Absolute Percentage Error for Training and Validation Dataset

MAPE TST : Mean Absolute Percentage Error for Testing Dataset

Table 4.2 Raw Data 1Day Lag –PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2121.25	19.35	2261.01	9.61
2	2121.16	19.20	2270.02	9.56
3	2121.76	19.03	2283.46	9.52
4	2121.47	19.08	2279.11	9.53
5	2126.26	19.09	2294.69	9.63
6	2121.54	19.36	2258.49	9.60
7	2121.77	19.11	2273.80	9.52
8	2121.22	19.15	2274.73	9.55
9	2122.09	19.33	2268.22	9.64
10	8718.40	50.41	9973.21	55.90

Table 4.3 Raw Data 1Day Lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	11869.02	52.52	12969.96	61.08
2	6859.26	74.89	5968.05	40.65
3	1459.06	9.37	1878.93	6.15
4	1310.13	3.91	1817.12	5.02
5	13629.06	55.88	14259.83	63.36
6	1189.07	3.70	1729.66	4.98
7	1195.89	3.81	1749.93	4.98
8	2607.34	3.65	2420.87	5.02
9	1185.71	3.50	1738.33	4.91
10	1200.98	3.52	1725.06	4.98

It is observed here that PURELIN architecture performs poorly in comparison with LOGSIG andTANSIG, both of which perform almost equally, but LOGSIG can be said to perform only marginally better.

As already stated in the selection criteria in last chapter, basing on the performance of Testing dataset, the lowest values of MAPE and RMSE consistent with each other are highlighted.

The results are graphically shown below for easy visualization.

In case of the training trial slow convergence is notable and for higher values of MAPE more than 4 epochs are sometimes required to achieve convergence. Each epoch typically consists of 1000 iterations. If convergence is not reached the network gives errors more than 100% even upto 300% to 400 %. Then it is required to increase the number of default number of epochs by using the slider as shown on the screenshot of Neural Network training (nntraintool) screen shot.

Completing of the training session by actual convergence due to reduction of error below the threshold is always preferable to stopping the training due to specified number of validation checks being performed successfully.

In the GUI the other performance parameter can also be observed by clicking the different options.

The spreads seen for this category of data are also not ideal as the spread width is large from the 45^0 line.

The difference about this shall be described in details in the discussion about Log-Transformed Datasets.

Especially to be noted are the random variation of both the error criteria in the raw datasets. Here also the better performance of the LOGSIG networks is observable in the plots as very rarely the error of LOGSIG networks varies randomly after reaching an optimum value. Whereas after decreasing continuously from one to 4 neurons, the MAPE and RMSE both suddenly increase for 5 neurons. This can be seen in the Fig. No.s 4.1,4.2, 4.3 and 4.4. same type of random variation is seen in case of PURELIN networks in many plots.

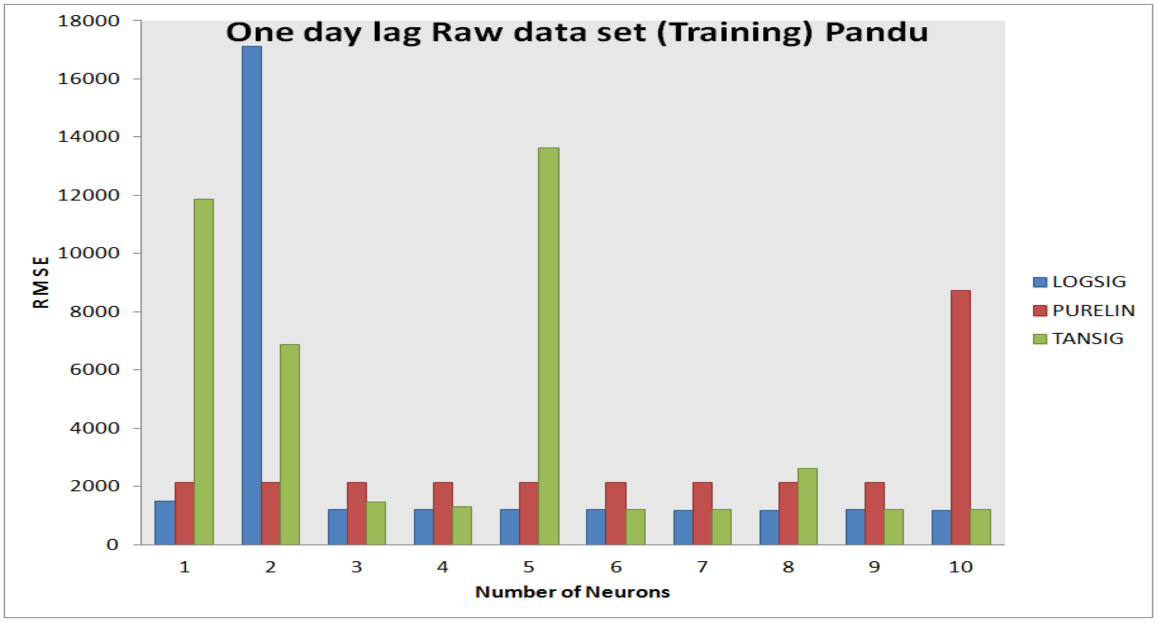


Figure 4.1 Raw Data 1 day lag RMSE TR (Pandu)

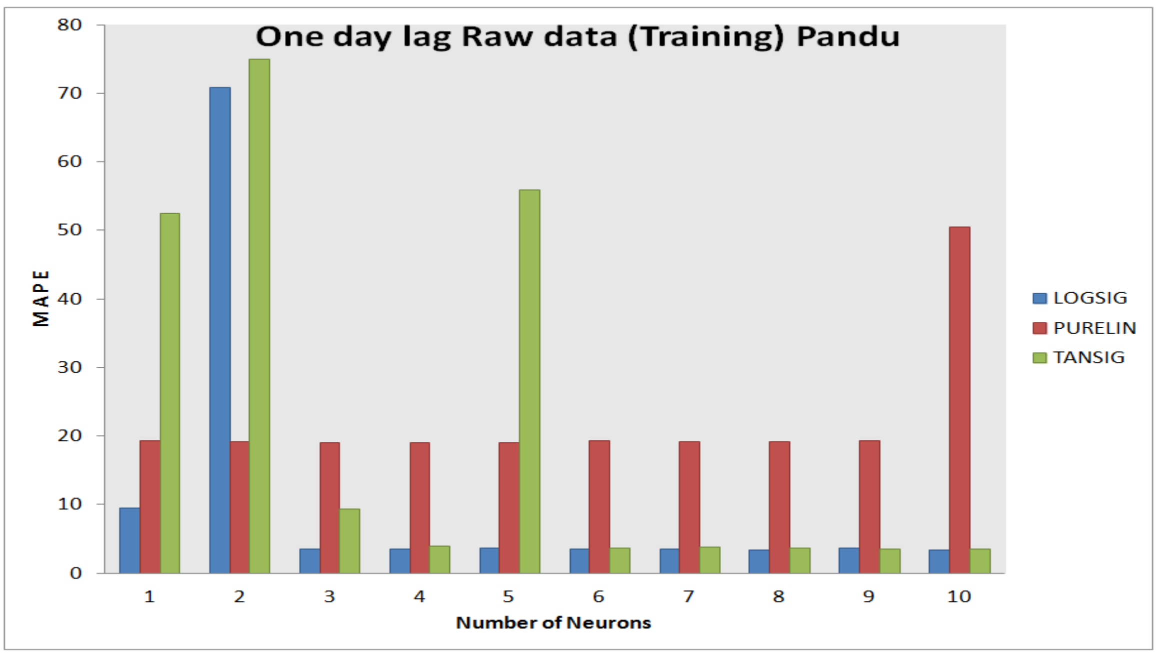


Figure 4.2 Raw Data 1day lag MAPE TR (Pandu)

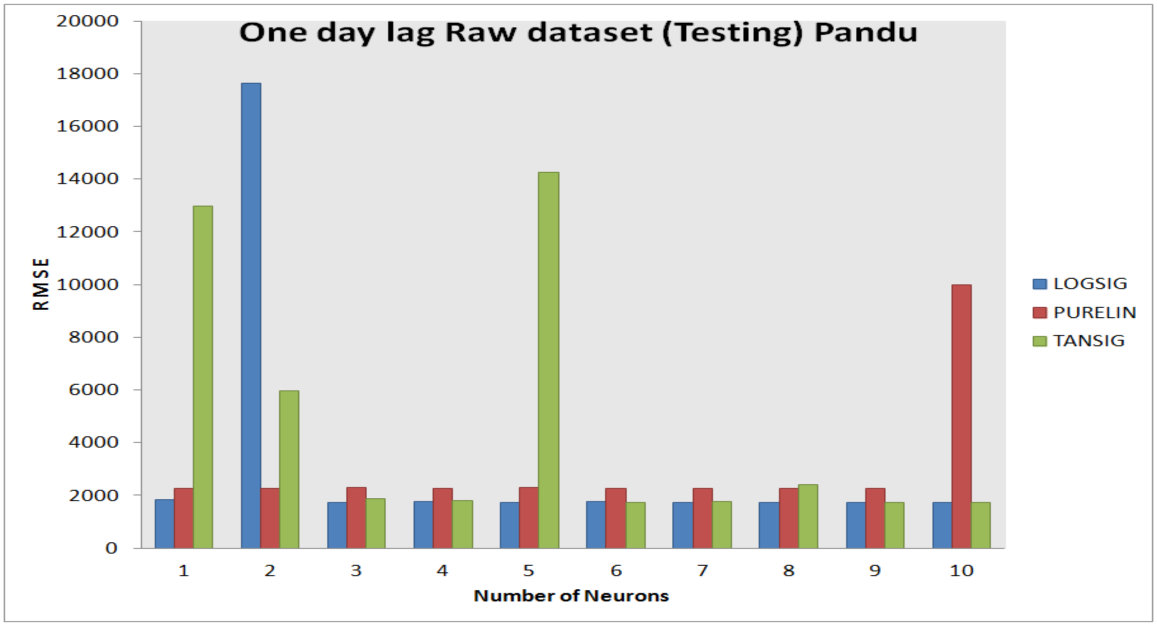


Figure 4.3 Raw Data 1day lag RMSE TST (Paandu)

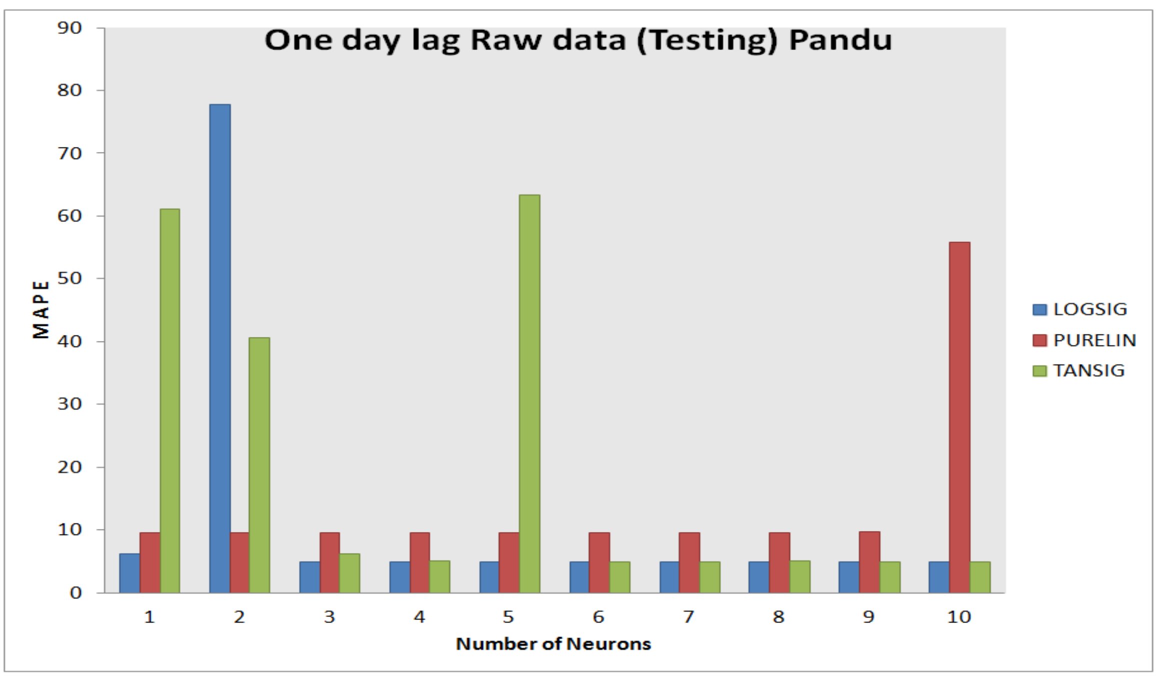


Figure 4.4 Raw Data 1 day lag MAPE TST (Pandu)

4.2.2 Raw Data Sets - Two Day Lag

Here the streamflow of two consecutive days is given as the input and the ANN predicts the streamflow of the next day. Thus the ANN has access to more information about the time series. The results of the performance evaluation of the 30 networks formed with these data sets is shown in the tables below.

It is also shown in graphical form for visualization.It is observed that LOGSIG architecture out performs both the PURELIN and TANSIG architectures. PURELIN architscture doesnot give the least error bot its performance is observed to be more consistent whereas some networks of the LOGSIG and TANSIG architectures give low errors, some give very high errors giving rise to inconsistent performance. The lowest values of errors in each architecture are highlighted.

The value of minimum error is reduced due to two inputs as compared to the previous dataset where only single input was given to the ANN.

In table 4.4 the very high values of RMSE and MAPE can be observed. Here all the three functions can be observed tobehave randomly as seen in the four plots from Fig. 4.5 to Fig. 4.8. Still the minimum error has come down from 3.46 (see Table 4.1) to 3.32 (see Table 4.4) after providing two input nodes to the ANN instead of one. This type of decrease in the value of minimum error is observable in many tables as we increase the no. of inputs from 1(in tables for onr day lag) to 2 (two day lag) and also further decrease when the inputs increase to 3 (three day lag). This is possible because for say 10 neurons,(just for example) the no. of synaptic connections and hence the number of weights for one input is 10, for two inputs it is 20 and for three inputs it is 30 between the input and hidden layer. The continuous and dynamic alteration in the matrix of the synaptic weights is stored as information in the programming of the ANN during the process of training, and once trained, this weight matrix analyses the previously unseen data and predicts the result during testing as well as during validation. This reflects in the decrease in values of RMSE and MAPE in almost all the tables and plots as the inputs increase from 1 to 3.

Table 4.4 Raw Data 2 day lag - LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1373.87	9.50	1138.97	4.54
2	16182.19	68.61	16882.94	75.94
3	18544.35	71.95	17355.88	60.07
4	1830.70	3.75	1970.69	3.83
5	11815.71	55.74	13204.27	63.66
6	16298.14	66.54	16643.71	71.45
7	1190.64	3.32	1115.43	3.39
8	1023.70	4.47	1411.73	4.51
9	18635.92	76.62	19245.89	82.67
10	1178.51	3.93	1101.59	3.75

Table 4.5 Raw Data 2 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2057.56	19.24	1658.45	8.64
2	2057.56	19.13	1668.59	8.62
3	2058.22	19.22	1660.15	8.64
4	2057.81	19.20	1671.51	8.69
5	2057.55	19.29	1658.03	8.67
6	2058.69	19.21	1667.48	8.65
7	13101.48	55.59	13678.90	62.21
8	2058.00	19.19	1663.98	8.64
9	16105.43	71.07	16783.34	77.03
10	2057.58	19.22	1663.75	8.66

Table 4.6 Raw Data 2 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	9389.17	51.64	9307.15	34.68
2	1371.08	9.28	1143.96	4.50
3	1245.72	4.70	1121.78	3.81
4	18601.87	75.37	19125.28	81.66
5	16572.52	73.60	17369.61	79.87
6	16951.46	75.76	16989.09	55.85
7	18116.07	75.91	18488.26	81.57
8	16540.36	72.45	17176.00	71.02
9	18714.46	76.71	19245.89	82.67
10	18426.65	76.36	19114.43	82.50

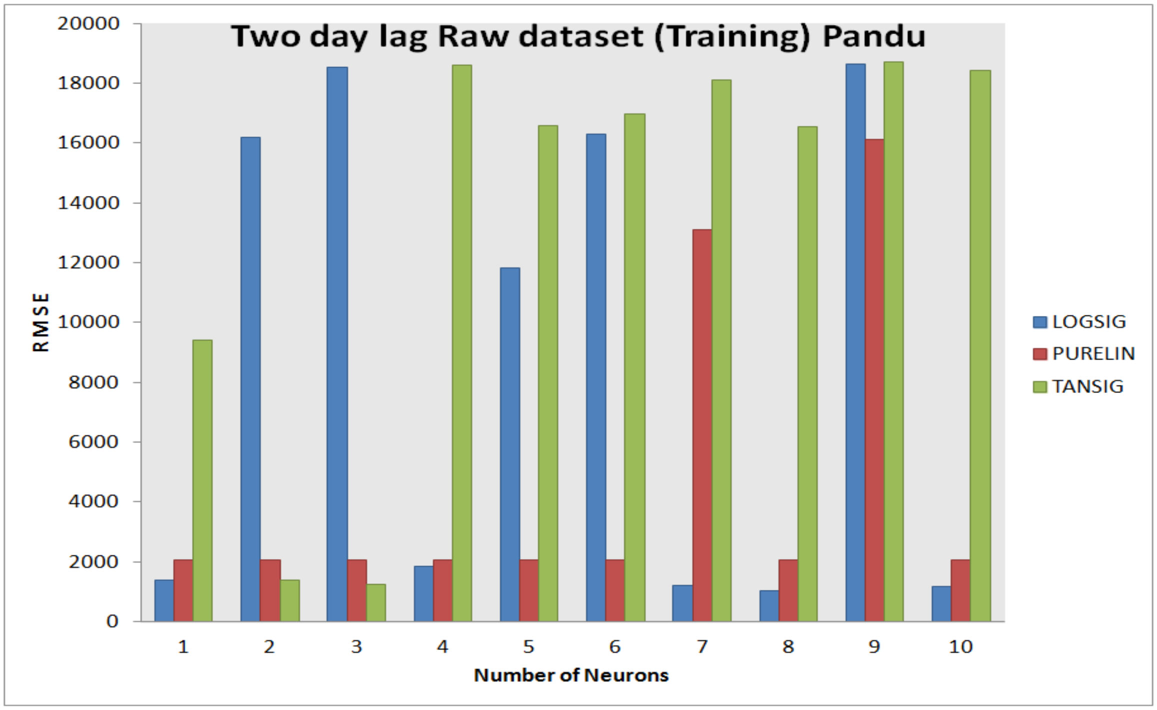


Fig. 4.5 Raw Data 2day lag RMSE TR (Pandu)

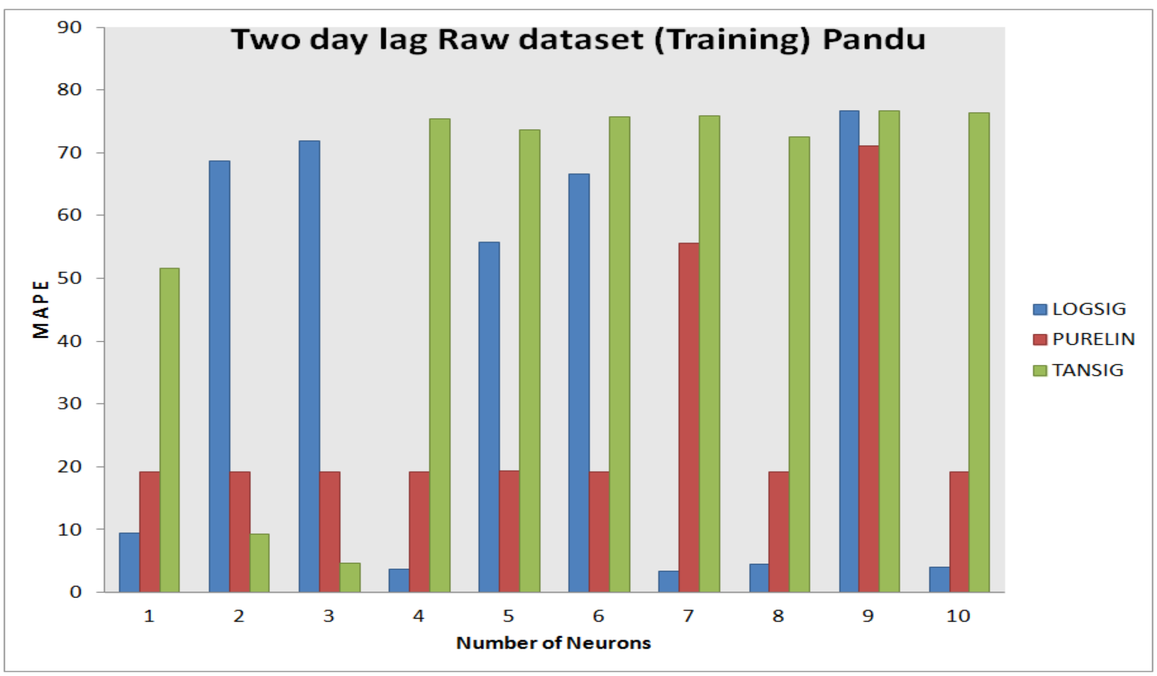


Fig. 4.6 Raw Data 2 day lag MAPE TR (Pandu)

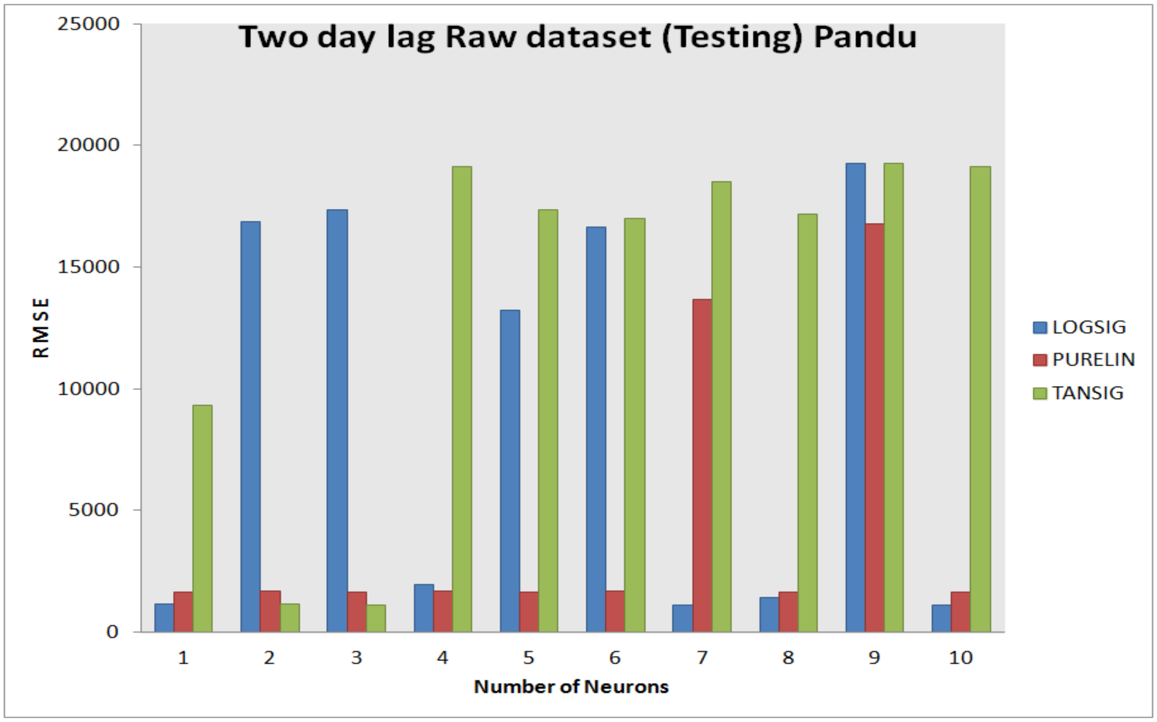


Fig. 4.7 Raw Data 2 day lag RMSE TST (Pandu)

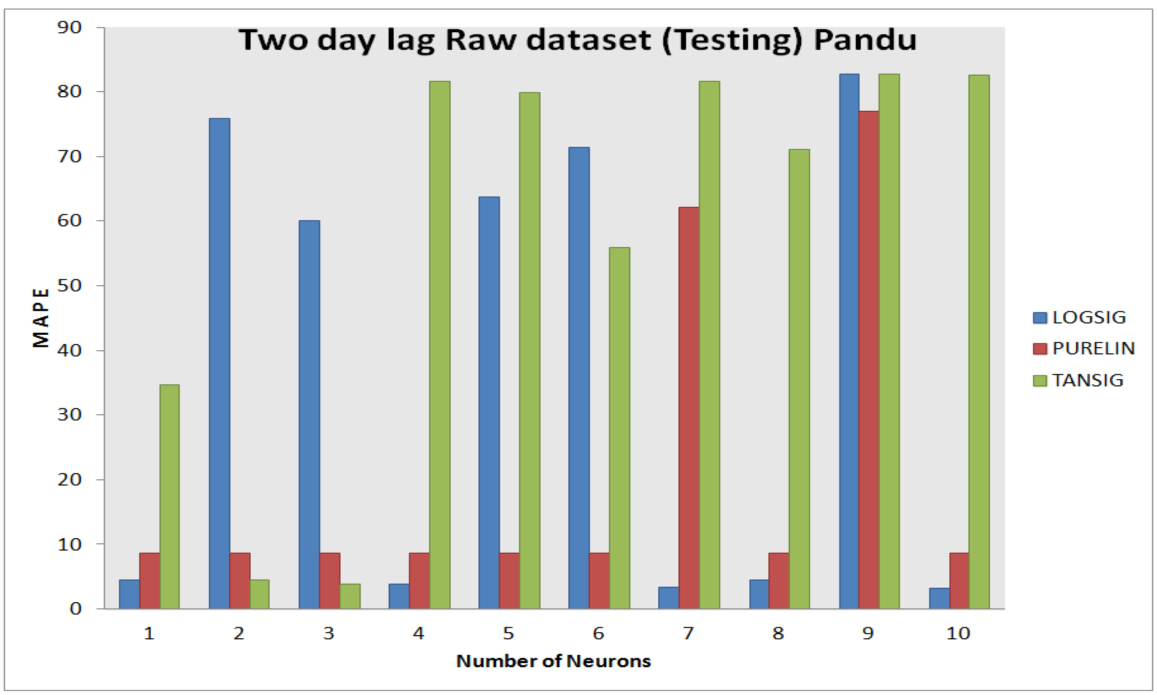


Fig.4.8 Raw Data 2 day lag MAPE TST (Pandu)

4.2.3 Raw Data Sets – Three Day Lag

Here the streamflow of three consecutive days is entered into the ANN as input and the predicted value of the streamflow for the next, i.e. the fourth day is obtained as the output. The results of the different ANN- Dataset combinations are shown in the table below.

Similar to the trend of previous datasets, here also non-consistent performance of LOGSIG and TANSIG architectures in comparison with PURELIN can be observed. Still, both of these architectures are able to give better prediction as indicated by very low values of bothy the RMSE and MAPE. Depending on the lowest MAPE values for the testing datasets consistent with RMSE and values of MAPE and RMSE for training and validation datasets are shown here by the cells with highlighting.

Three inputs further decrease the minimum error obtained as more information about the time series is available to the ANN.

Thus random variation of RMSE and MAPE seems to be the feature of all the plots for raw input data. Also high values of RMSE and MAPE are observed in the raw dataset. It becomes a drawback as it reflects upon the lack of robustness and dependability of the networks when dealing with previously unseen data during the testing trials.

All this can be observed and inferred from table 4.1 through table 4.8 and plots in fig. 4.1 through 4.12. Sometimes even contradictory behaviour of ANNs can be observed when error starts increasing after increase in the no. of neurons. This phenomenon is generally explained as the trapping into the local minima instead of reaching the global optimisation. But still many factors remain unknown about the internal working of the ANNs.

Table 4.7 Raw Data 3 day lag LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1373.47	9.51	1133.81	4.49
2	1151.46	3.45	1117.38	3.39
3	17562.75	70.09	19018.60	78.37
4	9324.82	45.50	10609.73	53.67
5	18460.17	76.39	19071.01	82.45
6	1408.39	3.88	1394.84	3.38
7	1210.52	3.14	1527.47	4.44
8	14373.32	46.87	14095.96	37.47
9	13768.76	64.93	14307.87	69.55
10	4387.28	5.53	3230.11	3.48

Table 4.8 Raw Data 3 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2050.86	19.09	1661.78	8.57
2	2053.04	19.15	1671.52	8.63
3	2056.61	19.19	1661.14	8.63
4	2050.29	19.13	1667.62	8.64
5	2050.64	19.00	1672.27	8.58
6	2052.16	19.02	1679.82	8.61
7	2051.53	19.32	1650.26	8.66
8	2050.57	19.05	1669.46	8.59
9	2050.73	19.09	1671.46	8.64
10	2051.82	18.99	1682.63	8.61

Table 4.9 Raw Data 3 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1373.00	9.24	1139.99	4.38
2	18724.85	76.72	19227.07	82.66
3	1144.51	3.80	1923.72	4.60
4	17703.81	64.12	18130.98	72.48
5	1063.26	3.99	1208.56	4.32
6	28020.72	265.17	18338.05	78.11
7	12335.16	62.12	13726.92	69.48
8	17478.69	74.84	17897.94	80.59
9	18820.90	77.30	19266.90	82.83
10	18089.23	75.83	18476.87	81.55

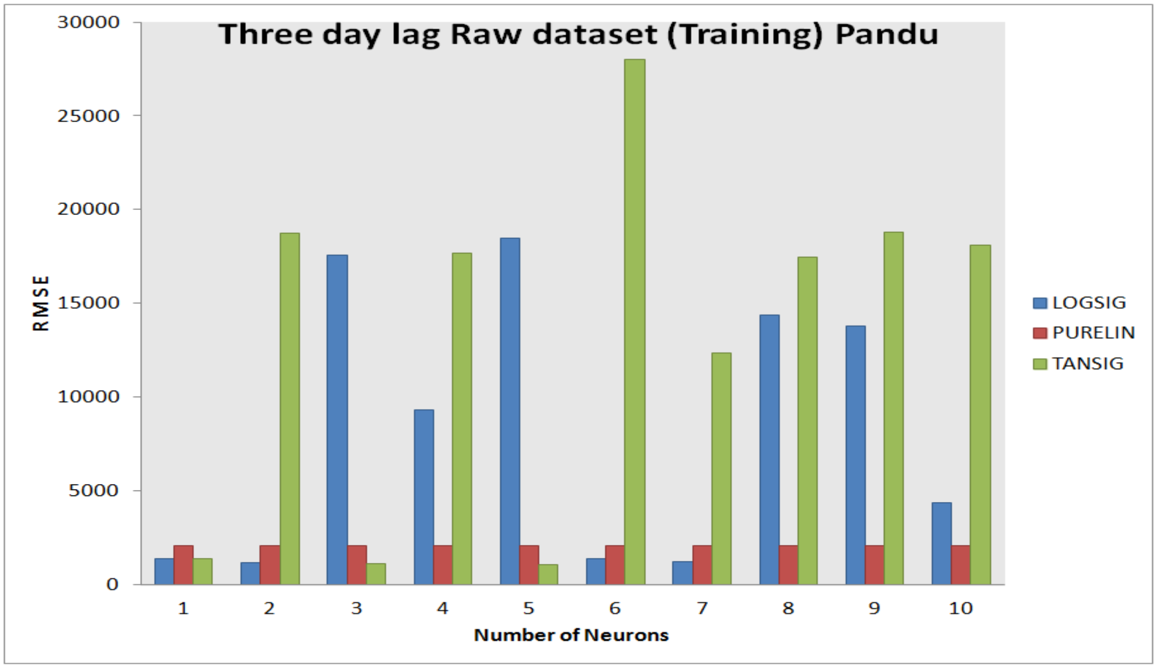


Fig. 4.9 Raw Data 3 day lag RMSE TR (Pandu)

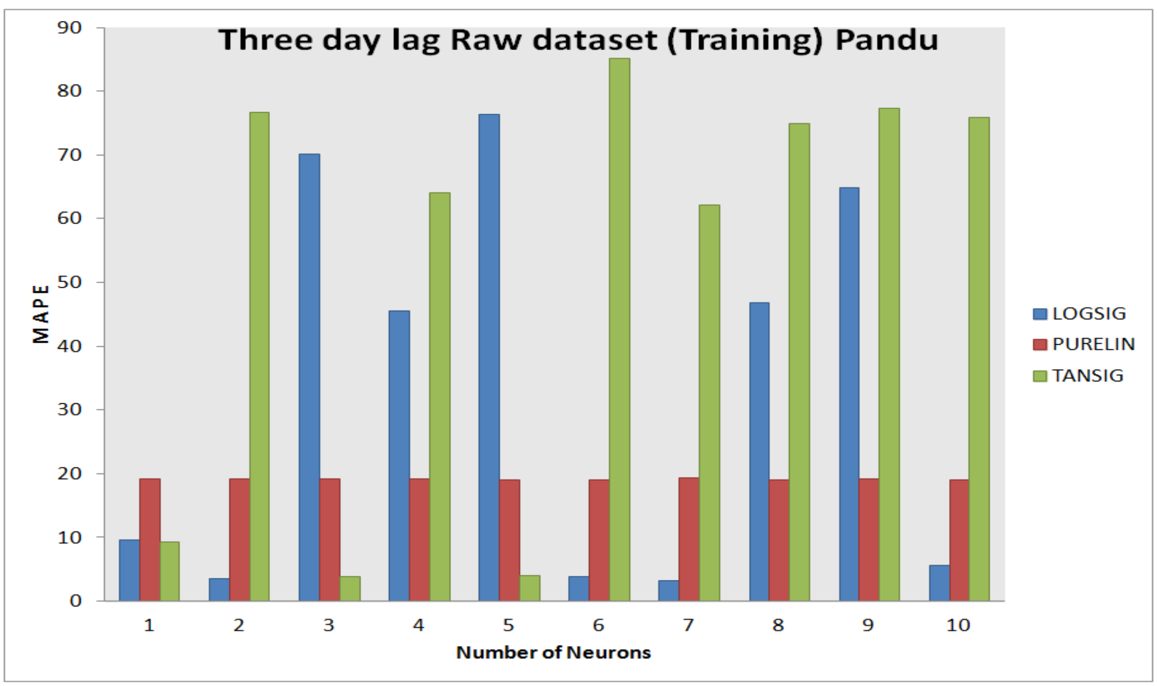


Fig. 4.10 Raw Data 3 day lag MAPE TR (Pandu)

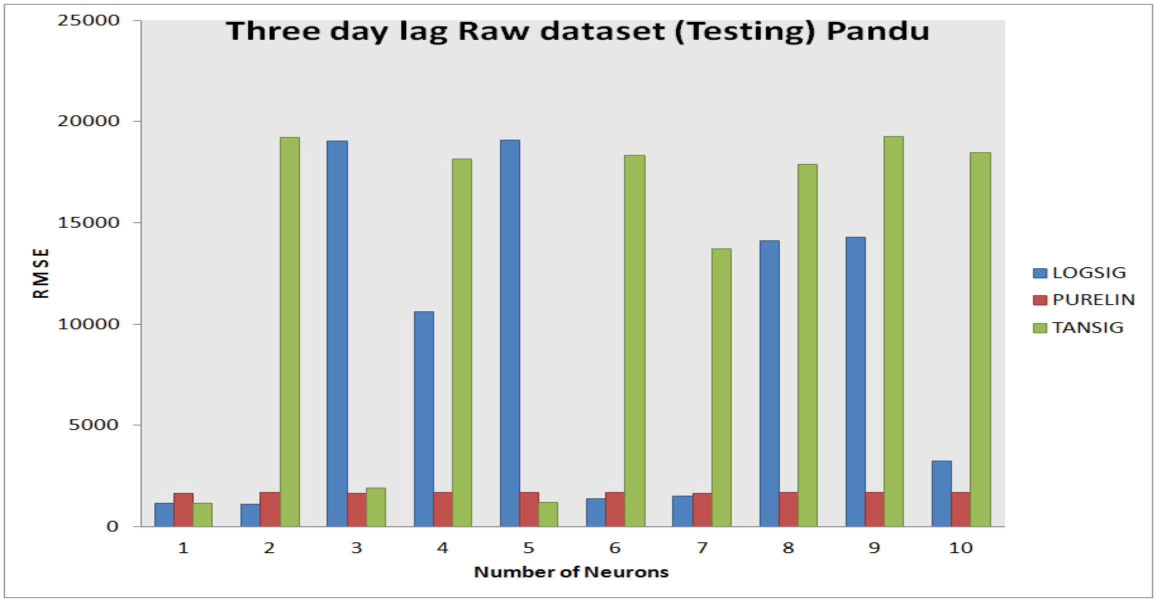


Fig. 4.11 Raw Data 3 day lag RMSE TST (Pandu)

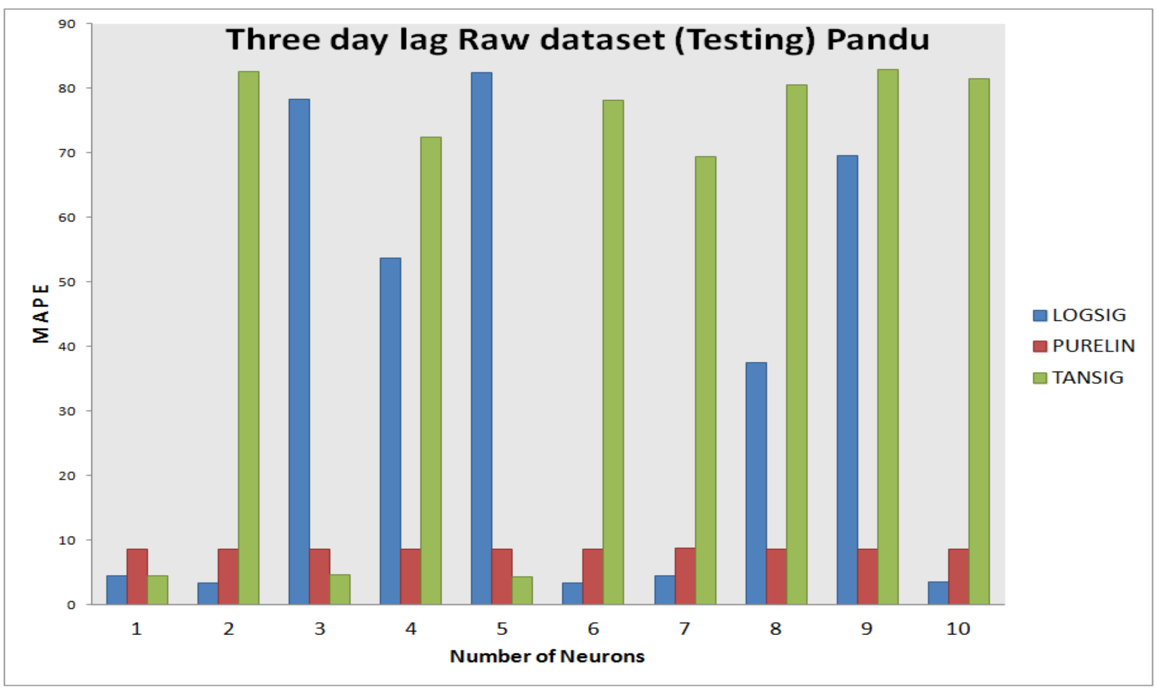


Fig. 4.12 Raw Data 3 day lag MAPE TST (Pandu)

4.3 Log Transformed Data

Here the logarithm of each data point is taken to the base 10 and the data are transformed by this pre-processing technique. Since the range of the data variation is very large in this particular situation of the Himalayan river, and the skewness coefficient is also high, logarithmic data is a logical choice of pre-processing technique as it will flatten the data spread into a much thinner band thus it may enable the ANN to perform better. Thus log-transform as a pre-processing technique may help in two ways:

1. It will flatten out the dataset which has very high peaks and low troughs
2. As the nodes essentially depend on summation of incoming data, the unknown method of formation of a complex matrix of synaptic weights may be facilitated as logarthmic mathematics is closer to summation even when the original data may exhibit non-linear, non-stationary and polynomial nature of higher powers of the variable.

As the ANN is a black box, and how the matrix of synaptic weights is formed is not yet fully known; and since the logarithmic transform is one of the scarcely tried transformations; it is justified to use this pre-processing technique and analyze the results. The output obtained is again transformed back to the original form of the streamflow values measured in m^3/s and then compared with the actual values of the streamflow for evaluation of performance of various networks.

4.3.1 Log Data - One Day Lag

The table below shows the results of the trials performed on the ‘training and validation’ and ‘testing’ datasets using log-transform.From the results it is observed that log-transform has provided stability in the performance and the irregular or erratic nature of the results seems to be removed by this pre-processing technique. The minimum values of error based on testing dataset results as the primary deciding criterion, are shown by the corresponding rows in each architecture category by highlighting.

In addition, a few things can be observed from these results:

The LOGSIG and TANSIG architectures both outperform the PURELIN and the LOGSIG seems to be just marginally but definitely better than TANSIG for this particular dataset. The results are also illustrated in graphics.

Table 4.10 Log Data 1day lag - LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1611.52	5.12	1500.02	4.48
2	1221.37	3.38	1049.17	2.76
3	1212.18	3.32	1038.68	2.67
4	1189.71	3.29	1020.61	2.67
5	1209.06	3.35	1035.88	2.72
6	1187.20	3.30	1023.33	2.72
7	1188.61	3.31	1023.86	2.73
8	1185.88	3.31	1020.12	2.71
9	1184.38	3.35	1019.93	2.75
10	1192.51	3.31	1020.01	2.73

Table 4.11 Log Data 1 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1586.65	5.11	1474.69	4.47
2	1598.82	5.09	1478.73	4.43
3	1592.89	5.10	1479.15	4.41
4	1582.56	5.12	1473.88	4.50
5	1589.99	5.11	1477.61	4.44
6	1588.02	5.10	1473.78	4.47
7	1597.21	5.09	1481.41	4.39
8	1588.68	5.11	1475.91	4.45
9	1583.42	5.12	1472.42	4.50
10	1585.46	5.11	1473.92	4.48

Table 4.12 Log Data 1 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1601.37	5.13	1492.77	4.53
2	1213.96	3.40	1040.94	2.79
3	1230.65	3.43	1057.88	2.81
4	1184.98	3.36	1018.50	2.76
5	1224.27	3.42	1053.24	2.81
6	1185.37	3.32	1019.77	2.71
7	1184.03	3.29	1022.06	2.72
8	1197.68	3.35	1043.20	2.75
9	1185.60	3.29	1019.03	2.74
10	1192.17	3.36	1022.92	2.77

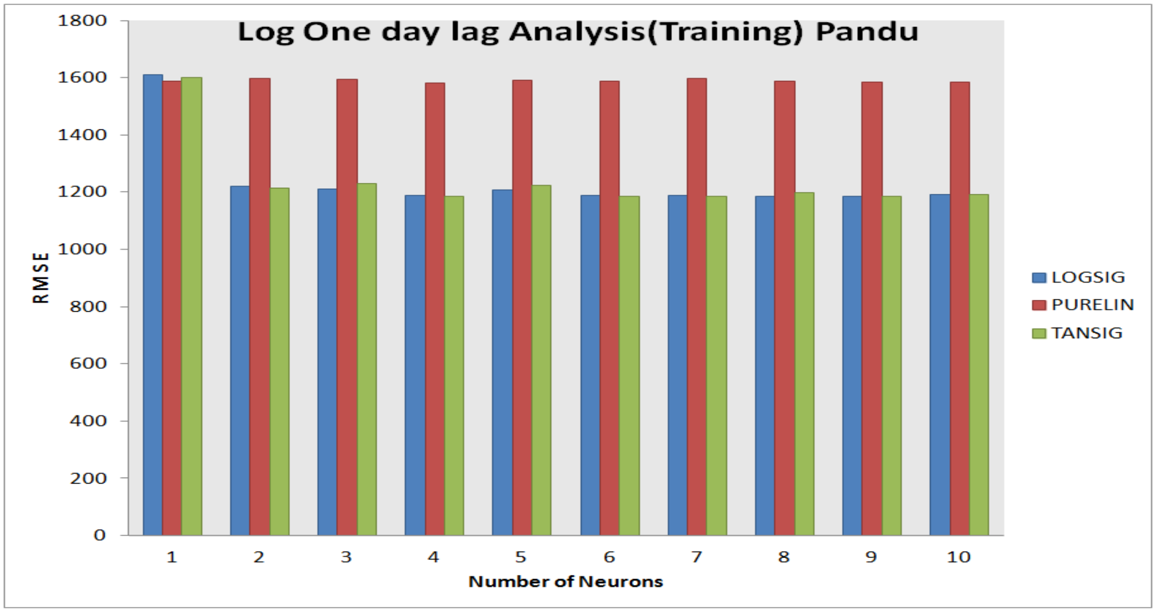


Fig. 4.13 Log Data 1 day lag – RMSE TR (Pandu)

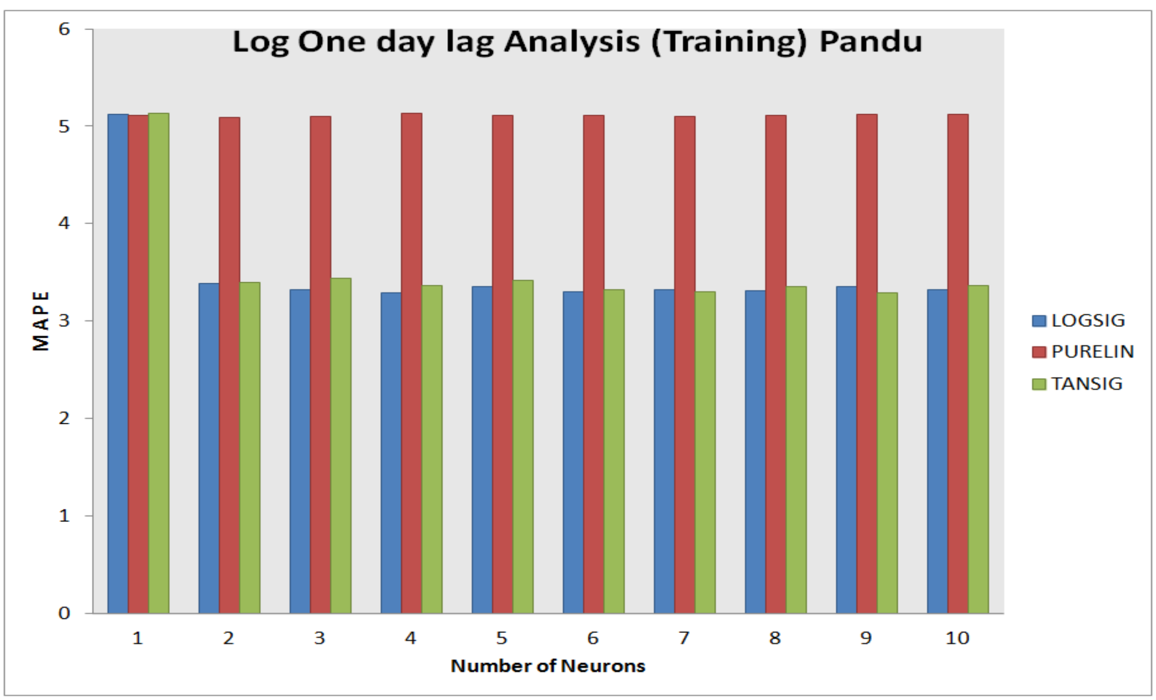


Fig. 4.14 Log Data 1 day lag – MAPE TR (Pandu)

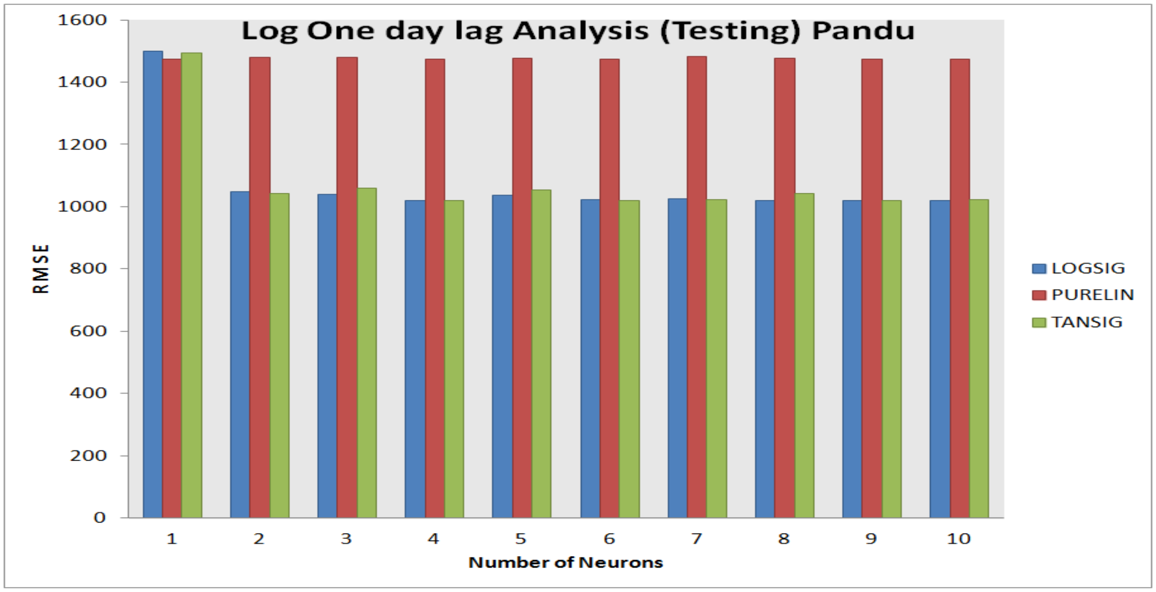


Fig. 4.15 Log Data 1 day lag RMSE TST (Pandu)

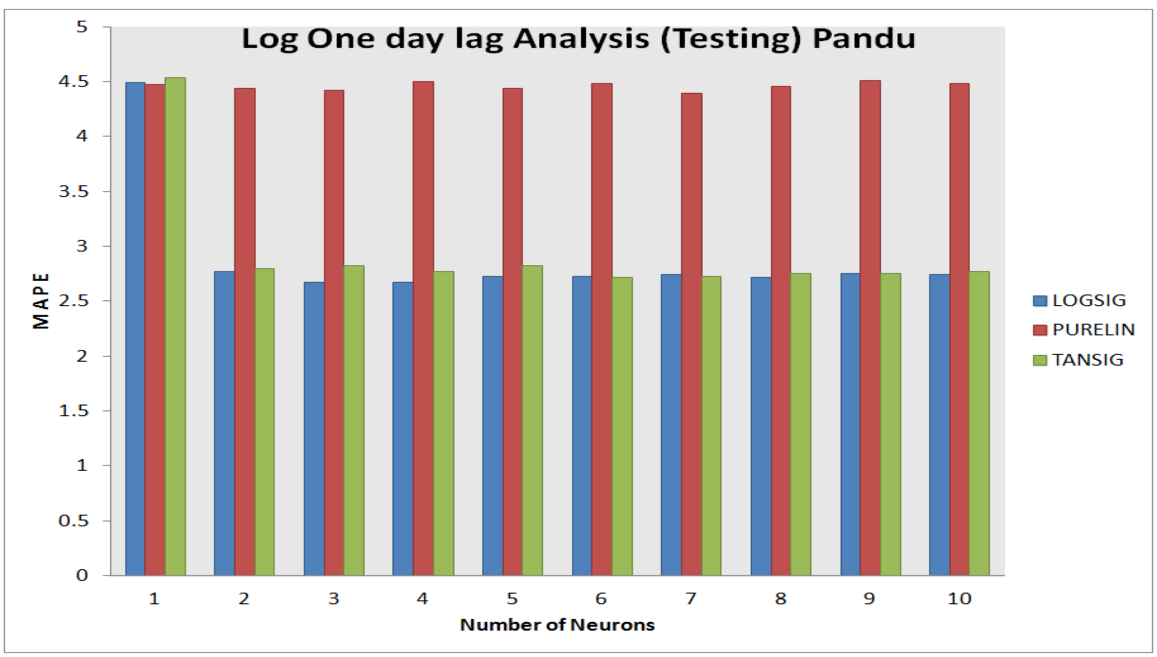


Fig. 4.15 Log Data 1 day lag – MAPE TST

4.3.2 Log Data – Two Day Lag

Here the log-transformed data of two consecutive days is given as input and the next day's streamflow is predicted. It is again re-transformed in the original units and then comapared with the original value. The results are represented in the observation table and by graphics following the observation table. The highlighting in the table shows the networks from each architecture giving optimum performance for this dataset according to the performance criteria adopted.

Here it is observed that the performance of PURELIN is poor compared to LOGSIG and TANSIG. In this dataset, LOGSIG and TANSIG perform almost equally, but the lowest value of error is given by TANSIG (MAPE = 2.11).

In the LOGSIG category, 2.17 is chosen in stead of 2.14, because the associated MAPE for training data for 2.14 MAPE is not consistent with it and also the associated RMSE has a higher value of 1461.394.

The same results are elaborated in the graphics.

While training the log transformed datasets many changes can be observed in the convergence pattern. In all trials the training is completed within one epoch. The convergence is quick and some trials converge to threshold minimum of error in less than 10 iterations. Stopping of the training by required number of validation validation checks is not observed and actual convergence by reduction of error below the threshold value occurs in all trials of this dataset.

Table 4.13 Log Data 2day lag - LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1485.48	4.76	1405.31	4.19
2	1070.15	2.81	943.52	2.22
3	1085.26	2.86	954.60	2.32
4	1032.69	2.71	909.22	2.17
5	1029.31	2.73	913.83	2.20
6	986.18	2.66	912.29	2.19
7	1452.79	3.81	1277.85	3.08
8	1045.46	2.82	924.87	2.26
9	1461.39	4.74	906.17	2.14
10	1051.77	2.75	916.45	2.23

Table 4.14 Log Data 2 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1468.46	4.75	1384.48	4.13
2	1471.63	4.72	1386.23	4.05
3	1464.98	4.75	1381.31	4.14
4	1467.76	4.75	1383.70	4.14
5	1478.21	4.75	1388.90	4.11
6	1460.83	4.74	1378.94	4.12
7	1461.19	4.74	1381.28	4.09
8	1469.00	4.80	1387.34	4.23
9	1461.39	4.74	1380.75	4.11
10	1464.62	4.73	1379.91	4.11

Table 4.15 Log Data 2 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1490.14	4.75	1407.07	4.13
2	1079.51	2.82	946.73	2.28
3	1066.99	2.78	908.12	2.22
4	996.86	2.68	946.19	2.23
5	1046.42	2.78	900.27	2.18
6	1052.02	2.75	918.13	2.32
7	982.64	2.63	927.72	2.24
8	985.21	2.67	890.29	2.11
9	985.21	2.67	902.55	2.17
10	1031.93	2.74	909.01	2.21

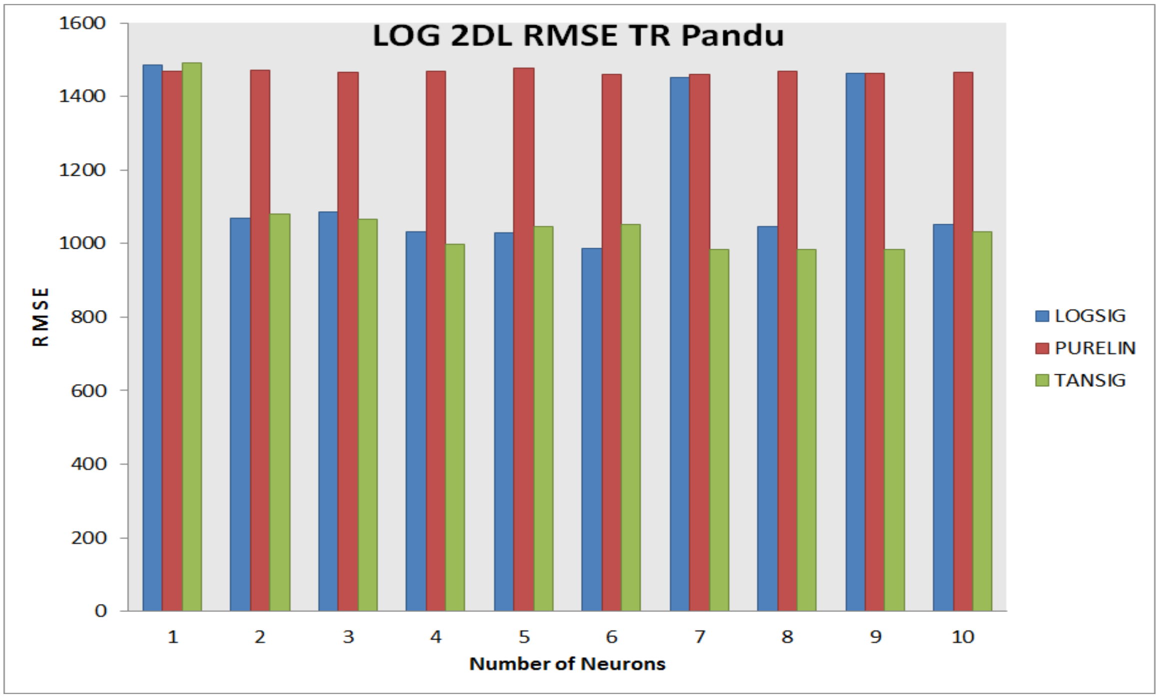


Fig. 4.17 Log Data 2 day lag RMSE TR (Pandu)

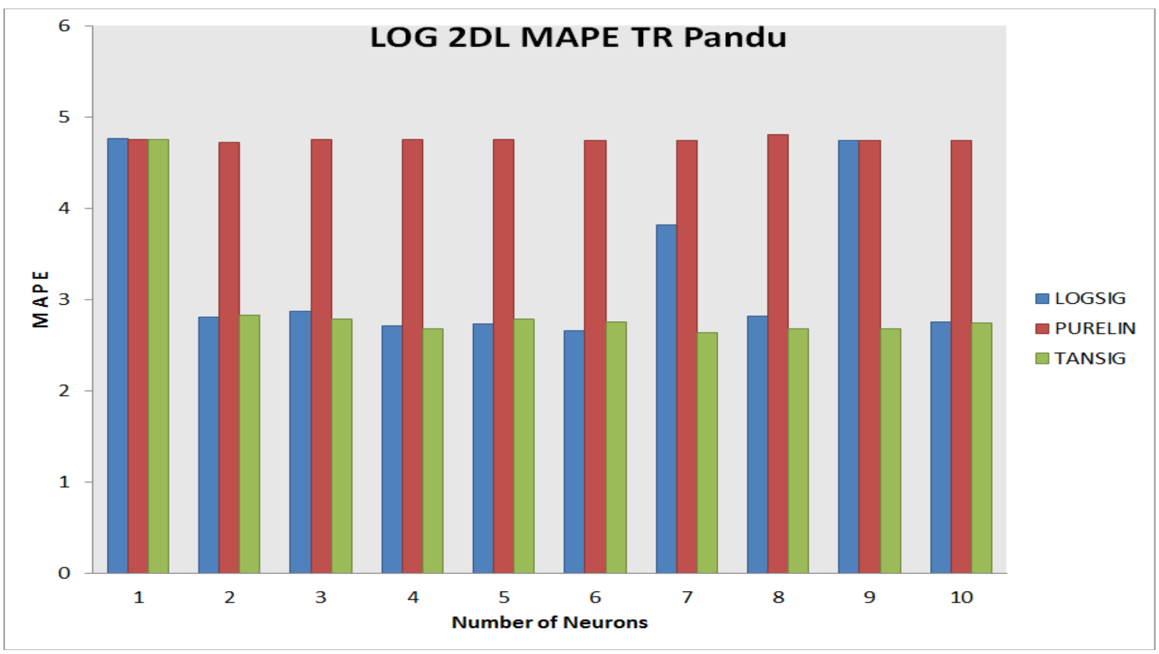


Fig. 4.18 Log Data 2 day lag MAPE TR (Pandu)

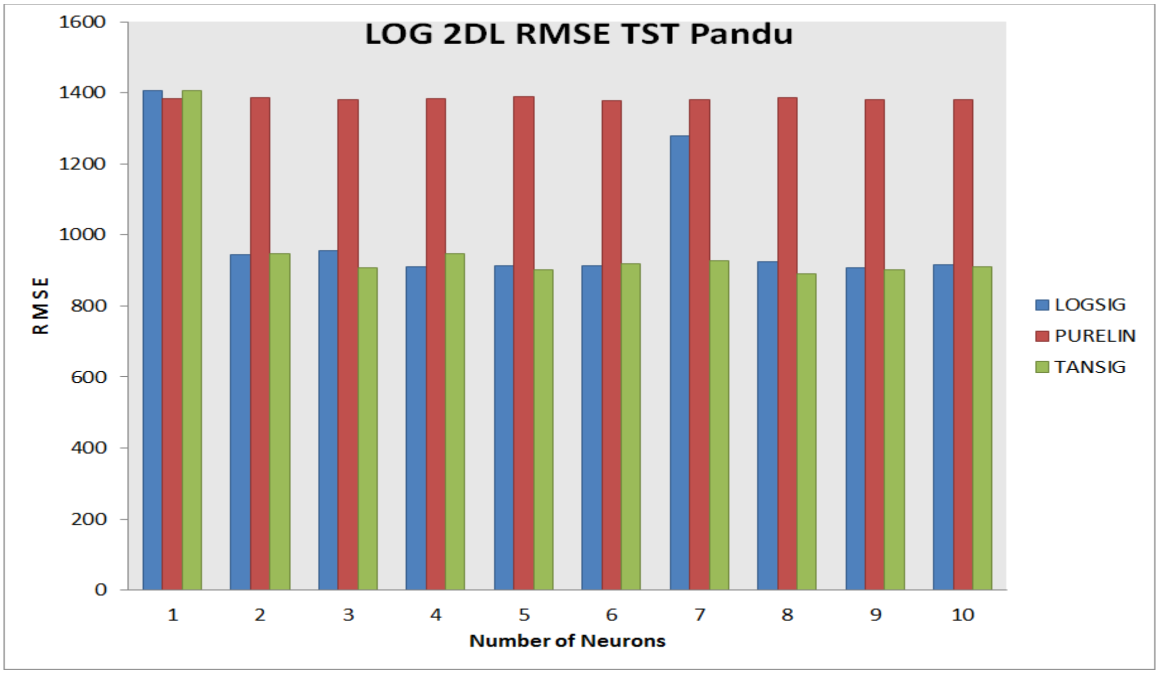


Fig. 4.19 Log Data 2 day lag RMSE TST (Pandu)

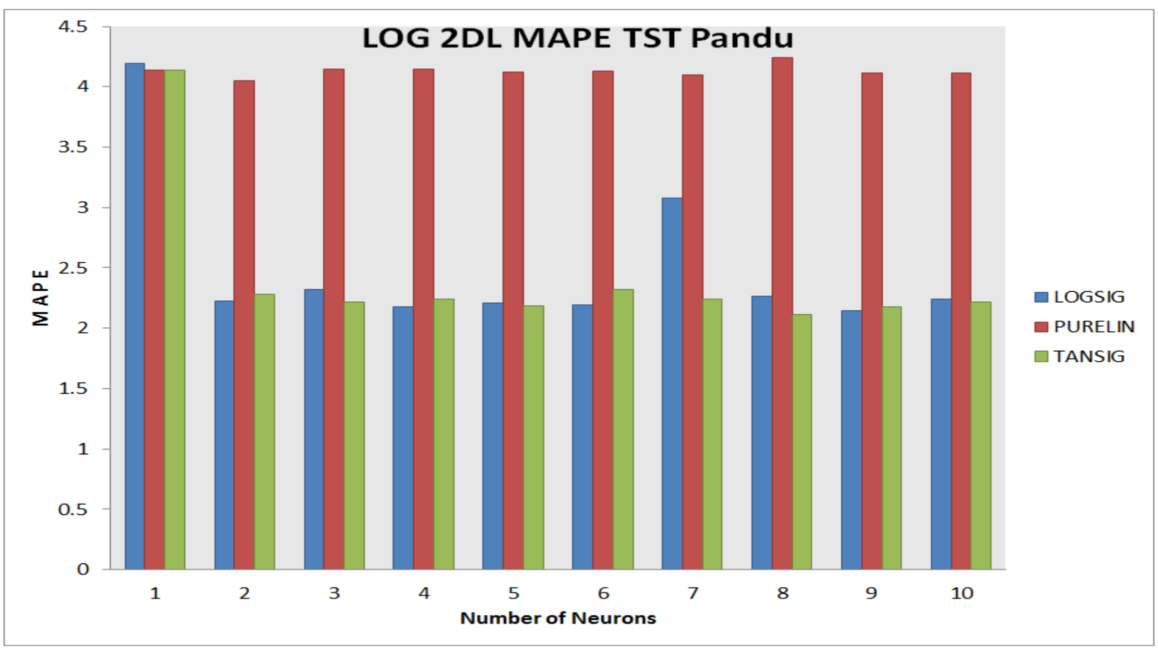


Fig. 4.20 Log Data 2 day lag MAPE TST (Pandu)

4.3.3 Log Data – Three Day Lag

Here the streamflow values are transformed in log to the base 10 and dataset for input contains data for three consecutive days, the value for the next day being predicted by each network. These values are re- transformed into original units and then compared with the observed values for calculating RMSE and MAPE for both Training-Validation as well as Testing datasets. The results are shown in the table below.

Here we see the best network performance by LOGSIG architecture, followed by TANSIG and PURELIN performs poorly in comparison with these two architectures. The network choice is shown by highlighting.

The performance is also represented by the graphics.

Log transform flattens the data distribution which is suitable to handle and non stationary nature of the given time series which is highly non linear and non stationary due to the particular situation of Himalayn rivers. The flow depends on the monsoons as well as on the freeze- thaw cycle of the snow. Both monsoons and snowmelt contribute to the build up of stream flow making the time series highly unpredictable in behavior if only raw data are used.

Looking at the results and comparing results for raw dataset and the log transformed datasets it is clearly seen that there is a marked increase in the performance as well as the stabilisation of randomness of resukts into consistent results for all the three architectures as the input dataset is changed from raw data to log transfornmed data.

For the combination of log data and PURELIN type networks the increase in the number of hidden neurons does not improve the performance significantly.

Table 4.16 Log Data 3 day lag - LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1475.85	4.76	1385.53	4.13
2	1076.27	2.87	951.26	2.32
3	1087.98	2.90	962.65	2.35
4	1045.26	2.76	939.67	2.25
5	1046.95	2.76	910.65	2.23
6	1002.80	2.67	907.95	2.12
7	1036.14	2.75	922.49	2.27
8	1002.56	2.66	900.17	2.13
9	1031.35	2.71	919.90	2.21
10	1475.85	4.76	1385.53	4.13

Table 4.17 Log Data 3 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1468.40	4.75	1378.16	4.11
2	1465.09	4.73	1374.49	4.07
3	1466.16	4.74	1374.84	4.09
4	1472.95	4.73	1381.04	4.06
5	1469.61	4.74	1375.60	4.08
6	1471.21	4.74	1379.97	4.09
7	1476.24	4.74	1379.54	4.10
8	1461.04	4.76	1375.35	4.17
9	1461.36	4.75	1374.94	4.13
10	1479.53	4.71	1382.93	4.01

Table 4.18Log Data 3 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	1484.22	4.74	1391.11	4.10
2	1074.33	2.82	945.98	2.28
3	1046.25	2.78	927.20	2.24
4	1055.01	2.75	937.61	2.23
5	1015.00	2.68	909.30	2.19
6	1031.30	2.73	913.66	2.21
7	1074.70	2.82	949.31	2.27
8	1042.68	2.74	930.89	2.22
9	1069.75	2.76	946.62	2.23
10	992.52	2.64	907.85	2.14

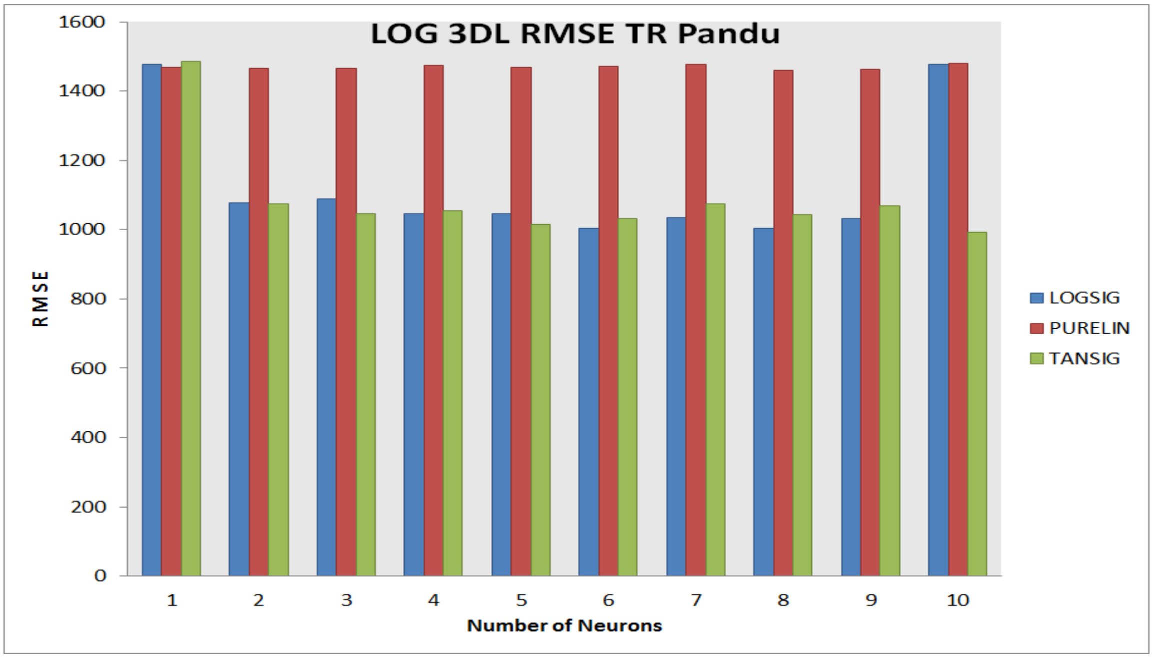


Fig. 4.21 Log Data 3 day lag RMSE TR(Pandu)

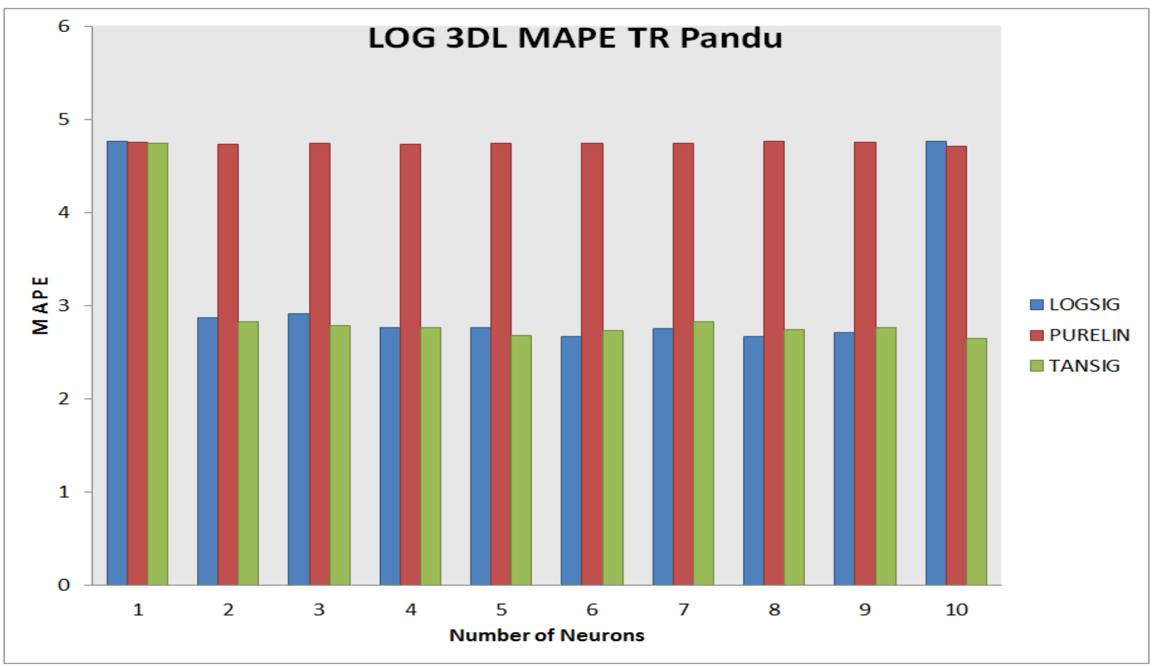


Fig. 4.22 Log Data 3 day lag MAPE TR (Pandu)

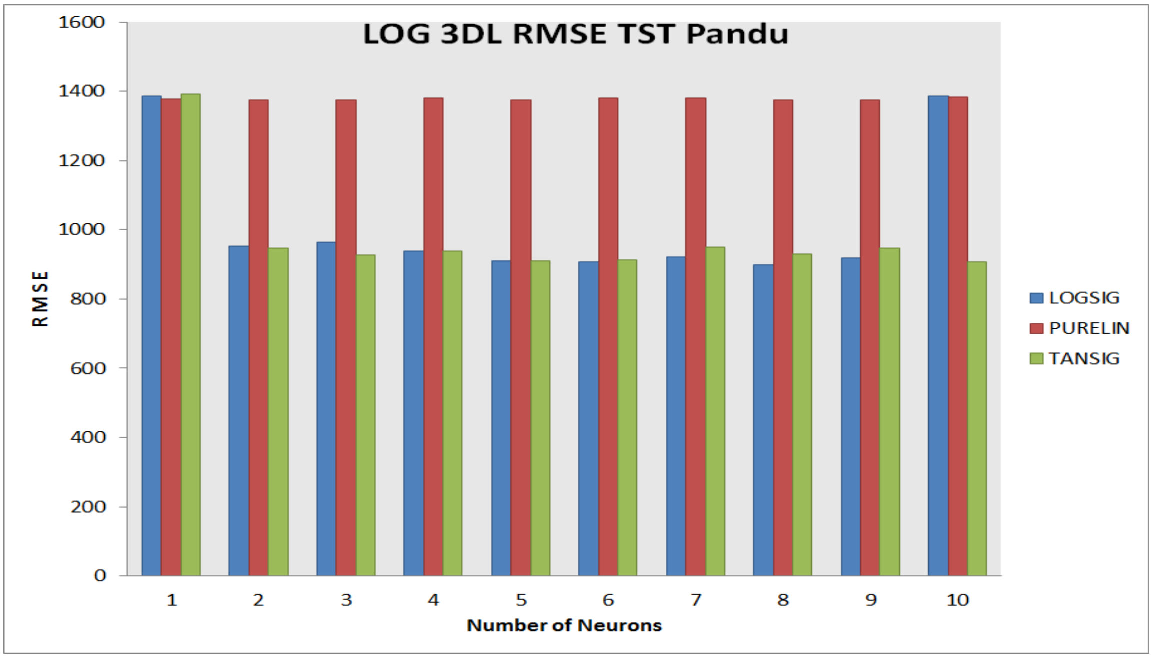


Fig. 4.23 Log Data 3 day lag RMSE TST (Pandu)

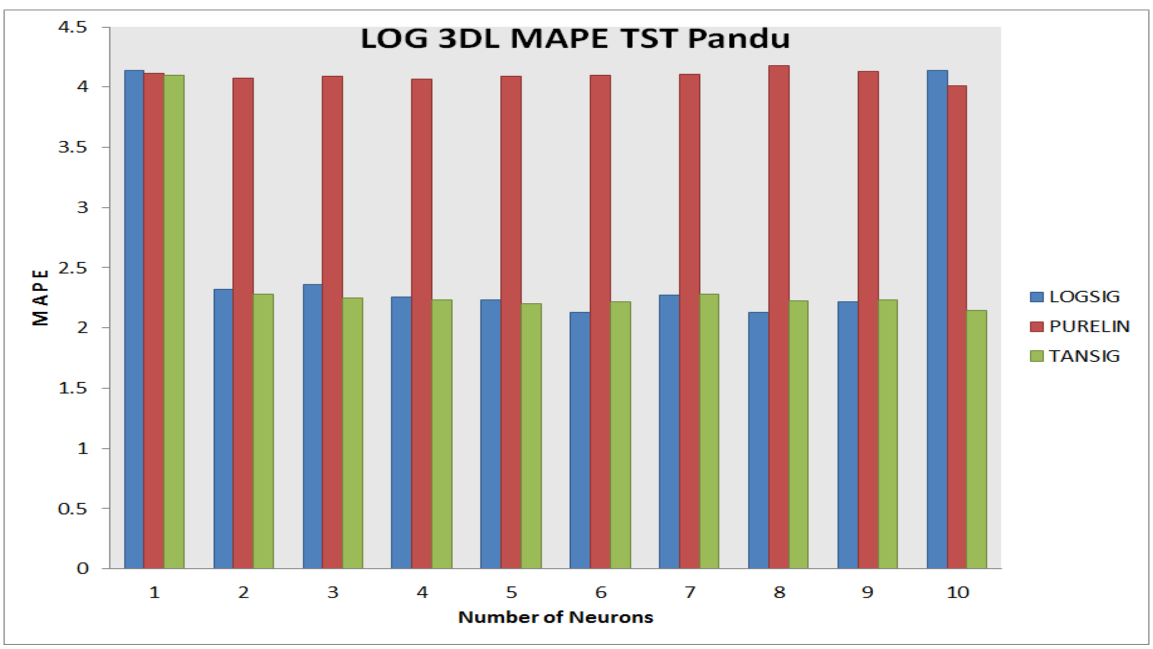


Fig. 4.24 Log Data 3 day lag MAPE TST (Pandu)

4.4 Log plus First Difference

Here the first difference, i.e. (x_i – x_{i-1}) is calculated after taking logarithm of each streamflow value and this first difference is added to the x_i value. For the very first value, i.e. for x_1, the first difference is taken as zero. Then this data are arranged in 1-day, 2-day and 3-day lag pattern and appropriate 2/3rd and 1/3rd data points are separated as training/validation set and testing set respectively.

4.4.1 Log plus First Difference – One day lag

After giving the one-day lag input, the results from various networks are re-transformed into corresponding original form and then compared for analysis and computations of error criteria.

The tables below show the result of this category.

The reuslts show that the overall performance is inferior as compared with the performance when log-transformed dataset is used. Overall, there is an increase in both the RMSE and MAPE of all the three categories. It is also observed that the discrepancy between the performance of PURELIN and LOGSIG/TANSIG is slightly reduced. The lowest error is given by the LOGSIG architecture. The lowest values in each category are shown by shading of the appropriate rows.

The results are also shown graphically.

The inclusion of first difference in the log plus FD dataset is supposed to remove any trends in the time series. But the results do not reflect upon increase of accuracy over log transformed datasets.

Table 4.19 Log+FD Data 1 day lag - LOGSIG

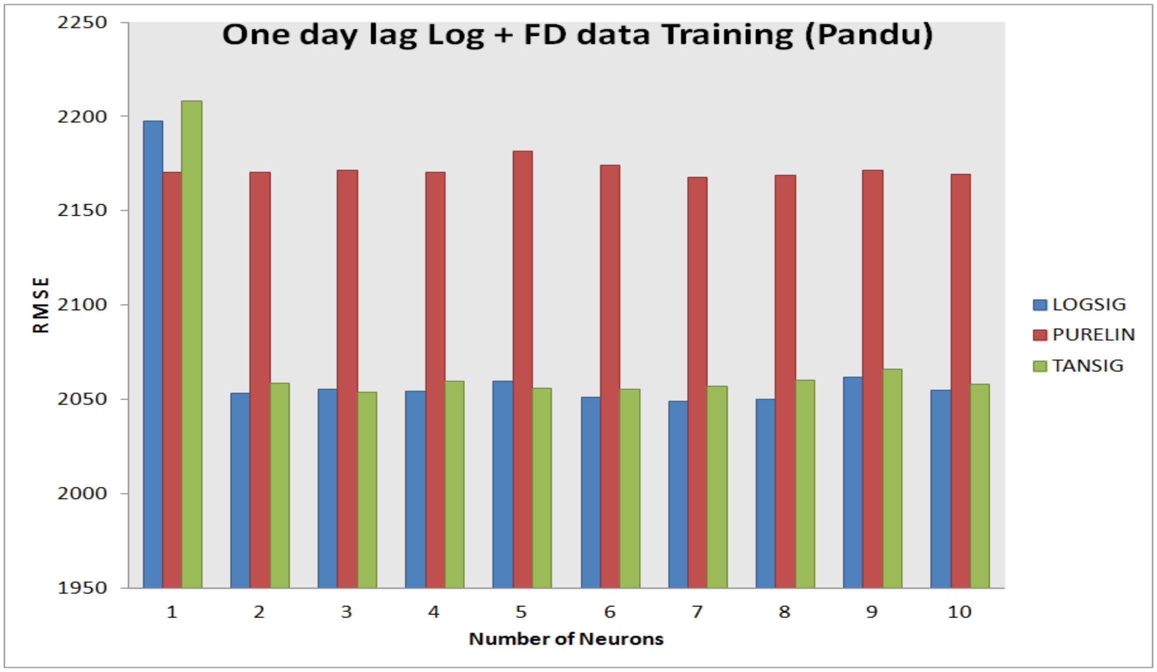
Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2197.80	6.48	1983.11	5.41
2	2053.54	5.45	1810.94	4.37
3	2055.57	5.42	1811.75	4.35
4	2054.15	5.41	1809.78	4.32
5	2059.91	5.52	1815.20	4.38
6	2050.96	5.48	1815.42	4.46
7	2049.15	5.50	1812.62	4.45
8	2049.92	5.45	1812.85	4.27
9	2061.77	5.49	1827.22	4.44
10	2054.88	5.49	1807.53	4.43

Table 4.20 Log + FD Data 1 day lag - PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2170.67	6.49	1955.34	5.36
2	2170.42	6.51	1957.00	5.39
3	2171.67	6.50	1956.93	5.35
4	2170.48	6.51	1957.01	5.39
5	2181.44	6.49	1962.19	5.20
6	2174.03	6.52	1960.12	5.33
7	2167.88	6.52	1956.13	5.46
8	2168.63	6.51	1956.57	5.44
9	2171.60	6.51	1958.40	5.37
10	2169.23	6.51	1956.38	5.42

Table 4.21 Log + FD Data 1 day lag TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2208.264	6.50	1994.32	5.37
2	2058.511	5.56	1816.55	4.55
3	2053.821	5.44	1810.21	4.37
4	2059.499	5.49	1818.14	4.44
5	2056.062	5.43	1812.62	4.35
6	2055.428	5.46	1811.75	4.36
7	2057.125	5.44	1812.06	4.33
8	2060.025	5.44	1816.48	4.37
9	2066.217	5.44	1815.57	4.41
10	2058.117	5.45	1821.19	4.38



Fig, 4.25 Log + FD Data 1 day lag RMSE TR (Pandu)

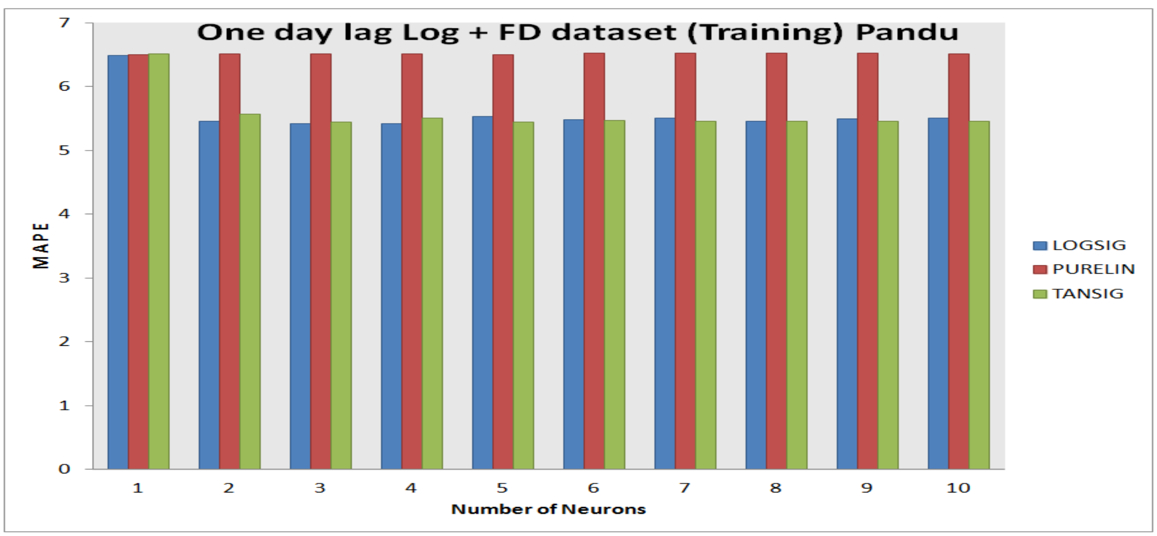


Fig. 4.26 Log + FD Data 1 day lag MAPE TR (Pandu)

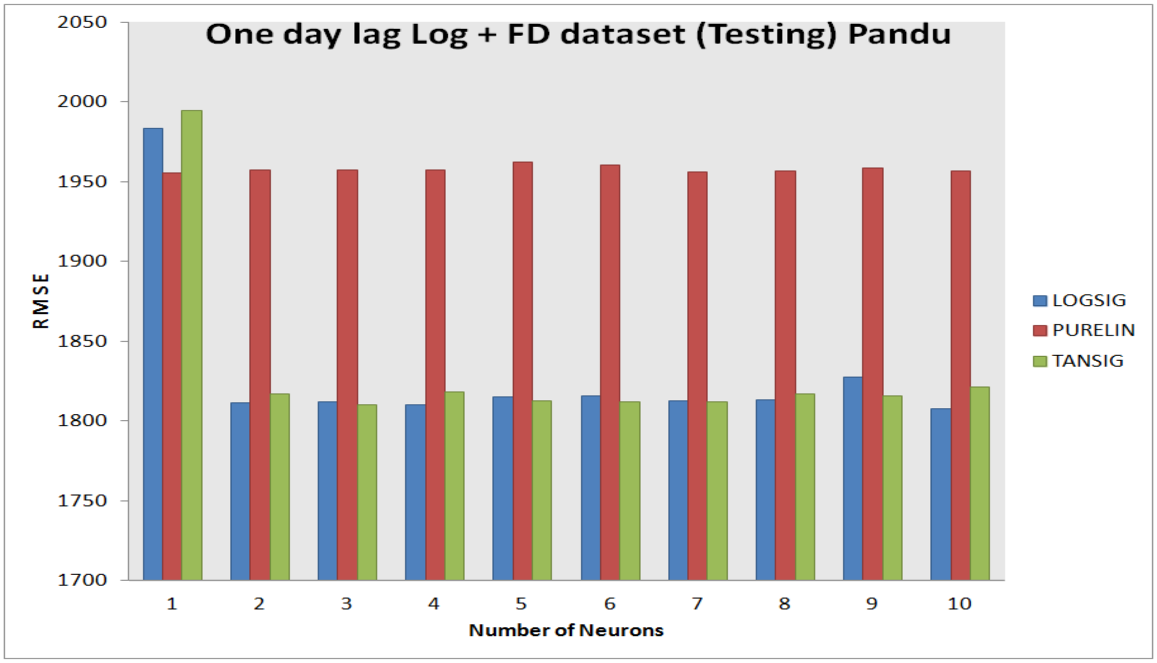


Fig. 4.27 Log + FD Data 1 day lag RMSE TST (Pandu)

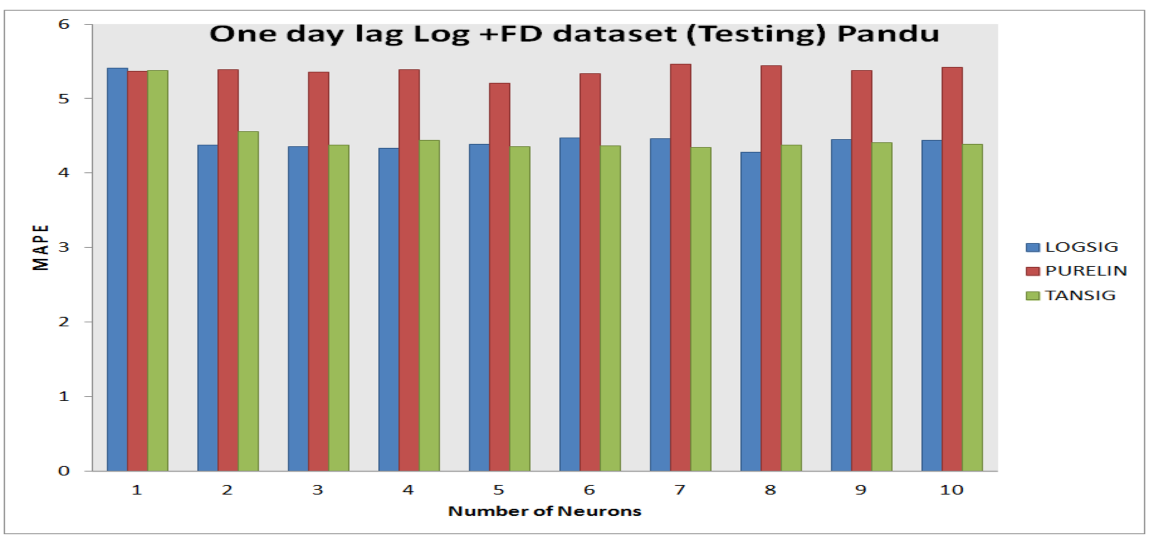


Fig. 4.28 Log + FD Data 1 day lag MAPE TST (Pandu)

4.4.2 Log plus First Difference – Two Day Lag

Here the input consists of log + FD data values of two consecutive days and the value for the next day is obtained as the output from the ANN. This value is reconverted to original form and compared with the actual streamflow value to evaluate the error and the required assessmant criteria.

The tables below show the results of these ANN trials and computations.

The result of this dataset also emphasizes a lower performance than the Log dataset. The PURELIN architecture remains the poorest performing one with LOGSIG and TANSIG being almost equal. Here the logsig (Testing MAPE 4.31) is the network of choice instead of the tansig network (Testing MAPE 4.30) as the RMSE values and also the training MAPE of the tansig network are much higher. This illustrates the overall policy in the choice of networks which considers consistency between the RMSE and MAPE values of both the sets of data in addition to the lowest MAPE of testing dataset as the preliminary criterion. The appropriate cells are shaded highlighting the lowest MAPE values.

These results are also represented in the plots.

Table 4.22 Log + FD Data 2 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2214.82	6.54	1985.45	5.43
2	2226.02	6.58	1998.13	5.53
3	2034.49	5.43	1808.94	4.33
4	2070.48	5.64	1923.52	4.71
5	1974.52	5.34	1783.65	4.36
6	1945.04	5.39	1814.41	4.45
7	2049.91	5.50	1835.12	4.41
8	2050.23	5.42	1828.25	4.31
9	2039.73	5.40	1812.58	4.35
10	1975.61	5.27	1793.76	4.31

Table 4.23 Log + FD Data 2 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2181.16	6.53	1951.95	5.45
2	2171.13	6.51	1952.18	5.42
3	2191.57	6.58	1960.97	5.41
4	2169.25	6.53	1957.12	5.43
5	2180.05	6.53	1952.39	5.44
6	2181.22	6.55	1957.12	5.38
7	2170.95	6.49	1950.77	5.36
8	2176.16	6.52	1951.46	5.41
9	2174.86	6.52	1951.78	5.44
10	2172.56	6.52	1953.57	5.42

Table 4.24 Log + FD Data 2 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2214.10	6.53	1985.90	5.47
2	2055.15	5.47	1814.82	4.42
3	2029.04	5.61	1820.99	4.40
4	2035.50	5.36	1818.27	4.30
5	2062.95	5.38	1831.02	4.31
6	1981.95	5.44	1813.34	4.48
7	1960.07	5.38	1827.30	4.38
8	2044.62	5.42	1828.71	4.40
9	1979.24	5.38	1798.37	4.39
10	2047.78	5.43	1808.24	4.42

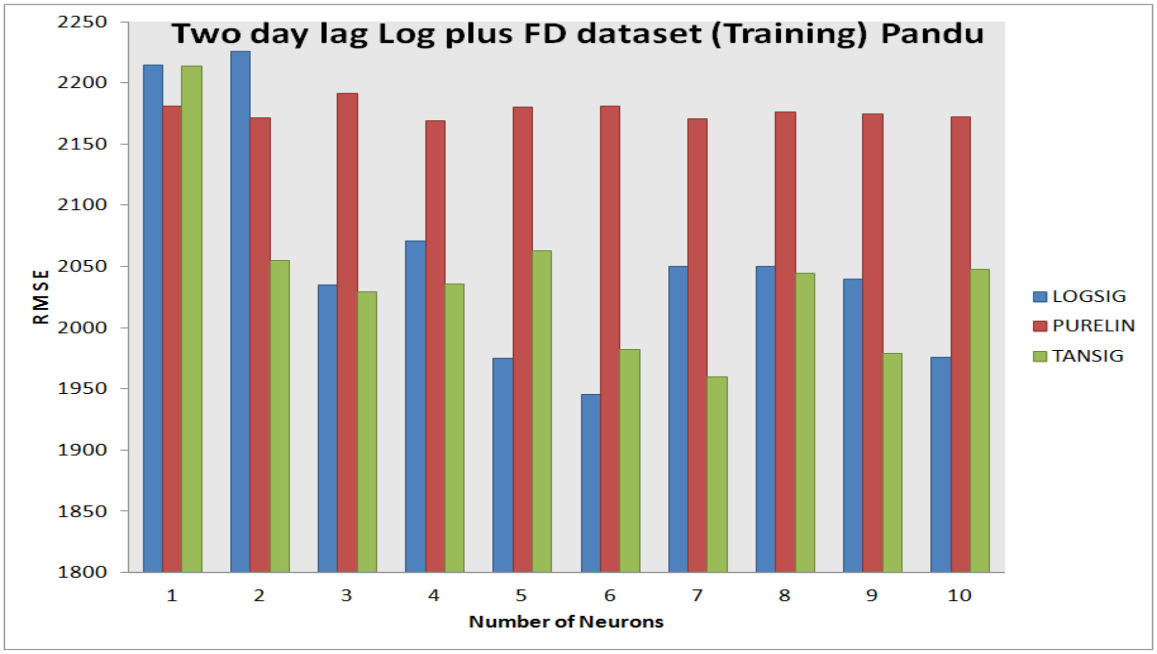


Fig. 4.29 Log + FD Data 2 day lag RMSE TR (Pandu)

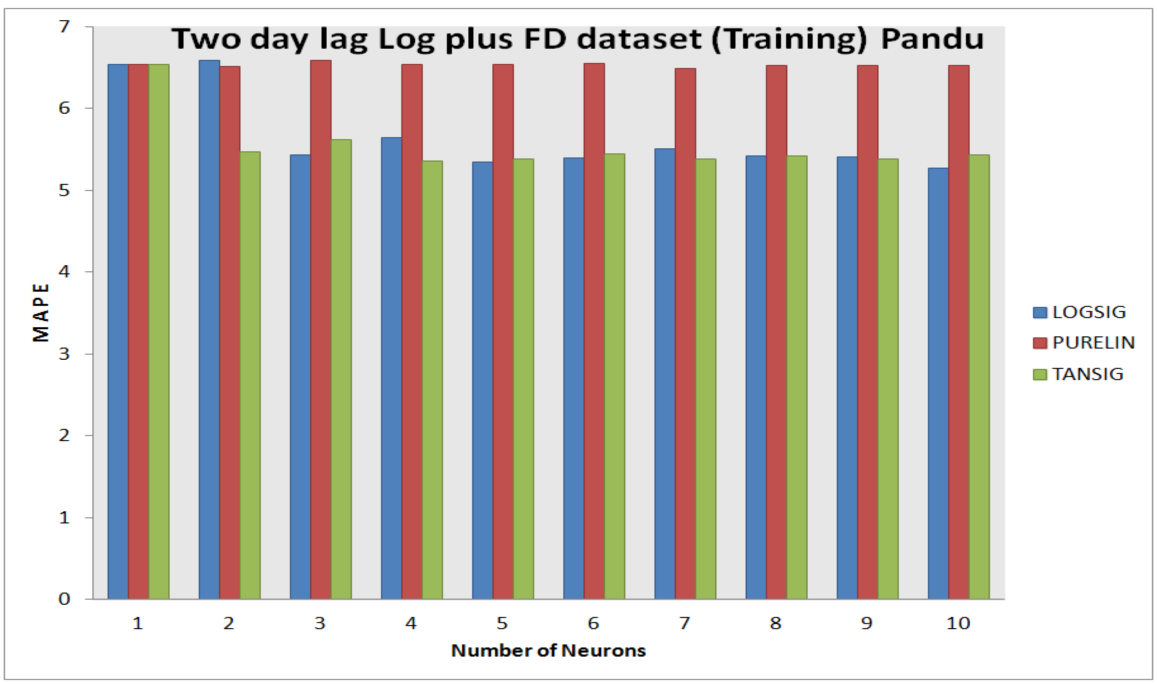


Fig. 4.30 Log + FD Data 2 day lag MAPE TR (Pandu)

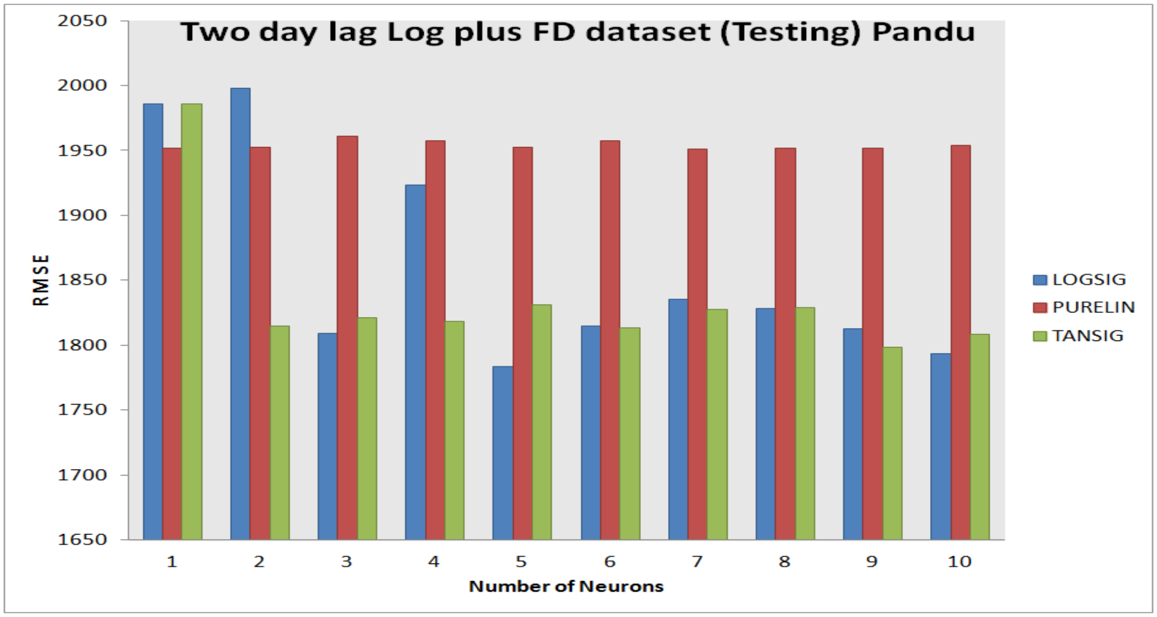


Fig. 4.31 Log + FD Data 2day lag RMSE TST (Pandu)

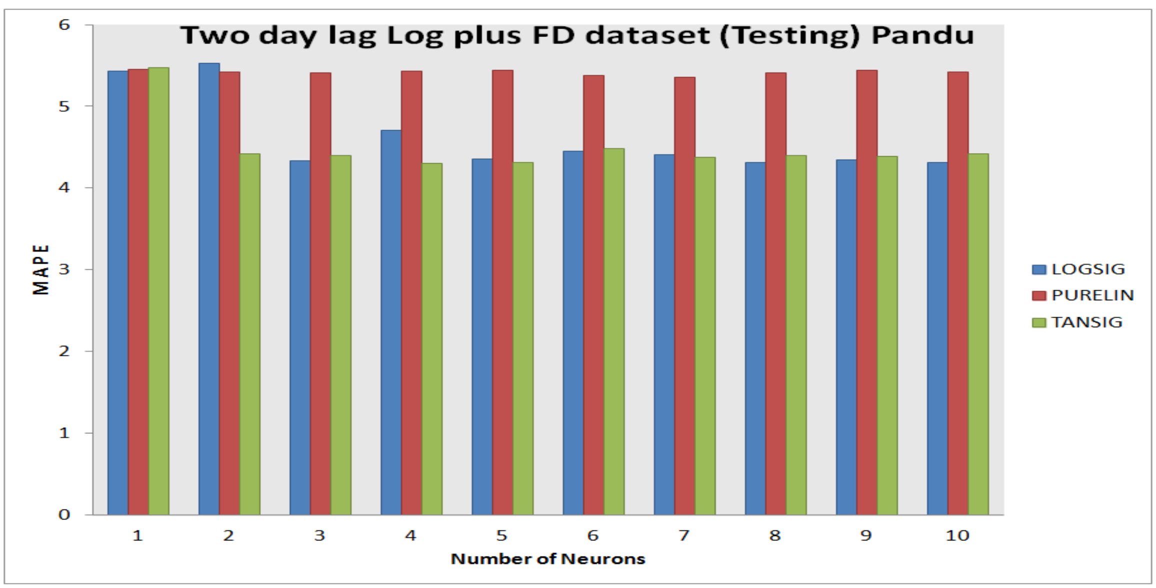


Fig. 4.32 Log + FD Data 2 day lag MAPE TST (Pandu)

4.4.3 Log plus First Difference – Three Day Lag

Here the data for three consecutive days from the pre-processed dataset of 'Log plus First Difference' is given to the ANNs as input and the predictions of the next day are obtained as output. The output is post-processed to bring it to original format and then these predictions of streamflow are compared with actual values to assess the evaluation criteria.

The performance here shows the same trend of PURELIN performing slightly lower than LOGSIG and TANSIG which perform almost equally well. The selected network in LOGSIG has testing dataset MAPE value 4.34 instead of the lowest value 4.32 as the corresponding RMSE values associated with 4.34 indicate a better choice against the mere marginal difference of 0.02 in MAPE values. The selection of most suitable network of each category is shown by shading.

The tables followed by graphics illustrate these results.

Logistic Sigmoidal activation function, i.e. LOGSIG, captures fully the variation of data behavior as compared to TANSIG which is a combination of hyperbolic tangent and sigmoidal function.

As seen in the previous chapter,(please refer to Fig. 3.8), the range of both TANSIG and PURELIN is between -1 and +1. Whereas the LOGSIG varies between 0 and +1 never reaching both the values but becoming asymptotically parallel to the time axis. How this may affect the ANN behavior is still a matter of conjecture, but the facts can be recorded from research works on ANNs dealing with time series.

Table 4.25 Log + FD Data 3 day lag LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2202.25	6.55	1972.20	5.47
2	2085.55	5.94	1854.05	4.72
3	2051.84	5.52	1817.47	4.47
4	2024.77	5.47	1792.77	4.41
5	2050.57	5.47	1822.84	4.38
6	1985.29	5.40	1793.80	4.35
7	2036.42	5.44	1826.54	4.41
8	1984.56	5.37	1774.85	4.34
9	2026.95	5.37	1826.18	4.32
10	2042.28	5.53	1847.30	4.45

Table 4.26 Log + FD Data 3 day lag PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2180.14	6.55	1953.85	5.45
2	2174.52	6.55	1954.51	5.39
3	2177.75	6.51	1947.61	5.36
4	2174.31	6.50	1950.67	5.37
5	2180.78	6.53	1952.01	5.37
6	2185.42	6.52	1949.18	5.42
7	2172.86	6.52	1953.17	5.43
8	2174.34	6.51	1951.85	5.30
9	2175.06	6.52	1946.67	5.45
10	2182.35	6.53	1951.41	5.38

Table 4.27 Log + FD Data 3 day lag TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2207.08	6.54	1979.05	5.45
2	2054.21	5.52	1813.29	4.41
3	2055.01	5.52	1822.52	4.42
4	2054.75	5.44	1808.93	4.38
5	2045.11	5.74	1809.02	4.42
6	1937.47	5.35	1827.98	4.36
7	2060.73	5.47	1815.97	4.44
8	2050.48	5.48	1863.13	4.49
9	1931.18	5.30	1797.64	4.36
10	1966.27	5.36	1836.48	4.43

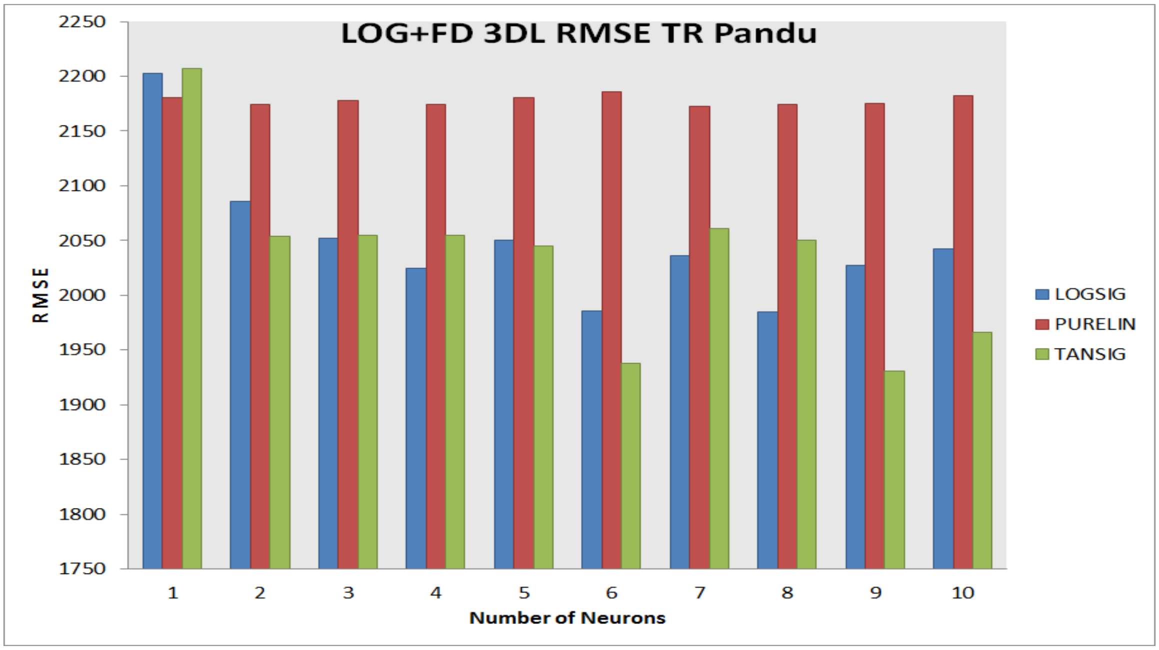


Fig. 4.33 Log + FD Data 3 day lag RMSE TR (Pandu)

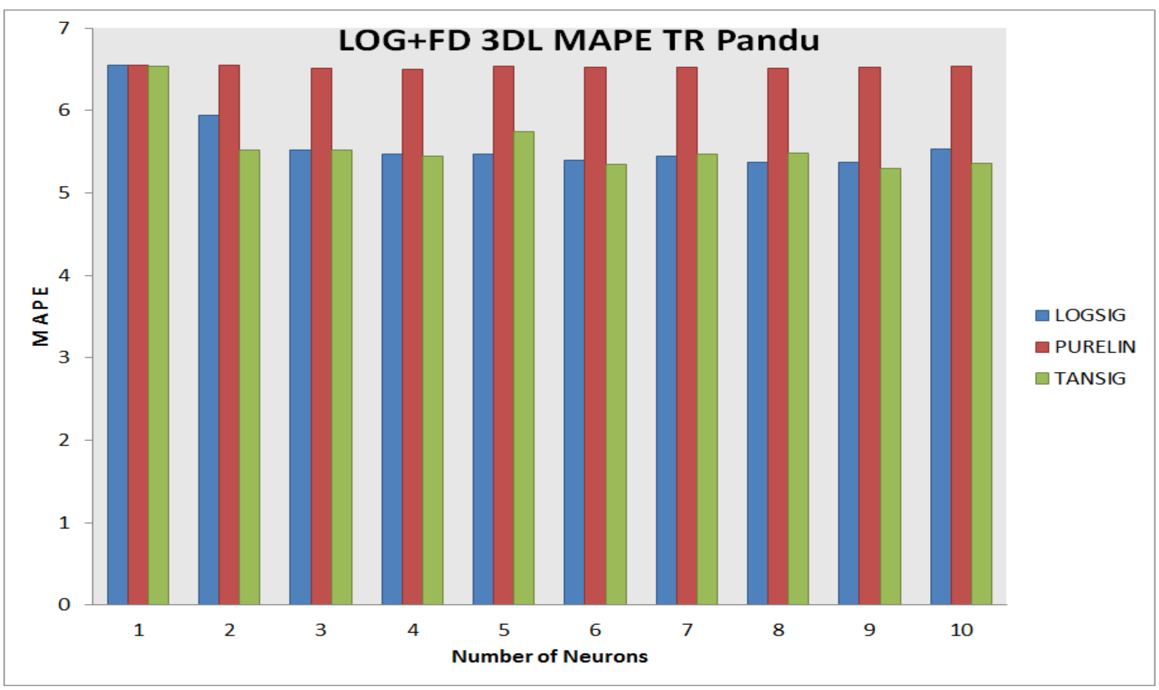


Fig. 4,34 Log + FD 3 day lag MAPE TR (Pandu)

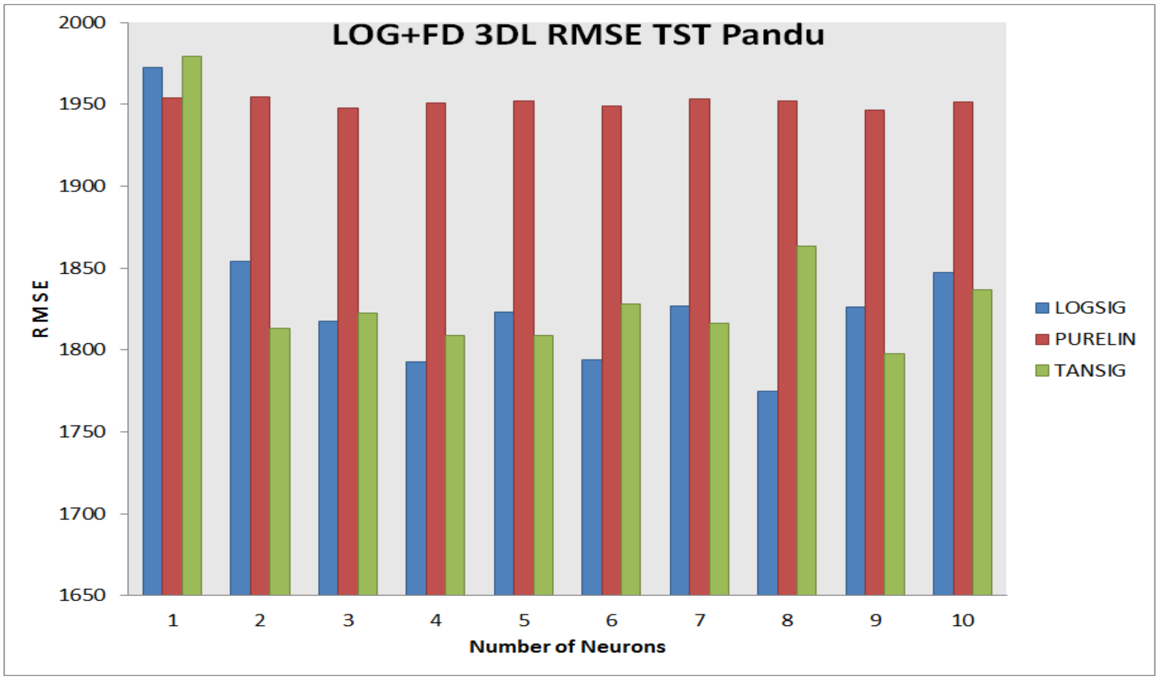


Fig. 4.35 Log + FD Data 3 day lag RMSE TST (Pandu)

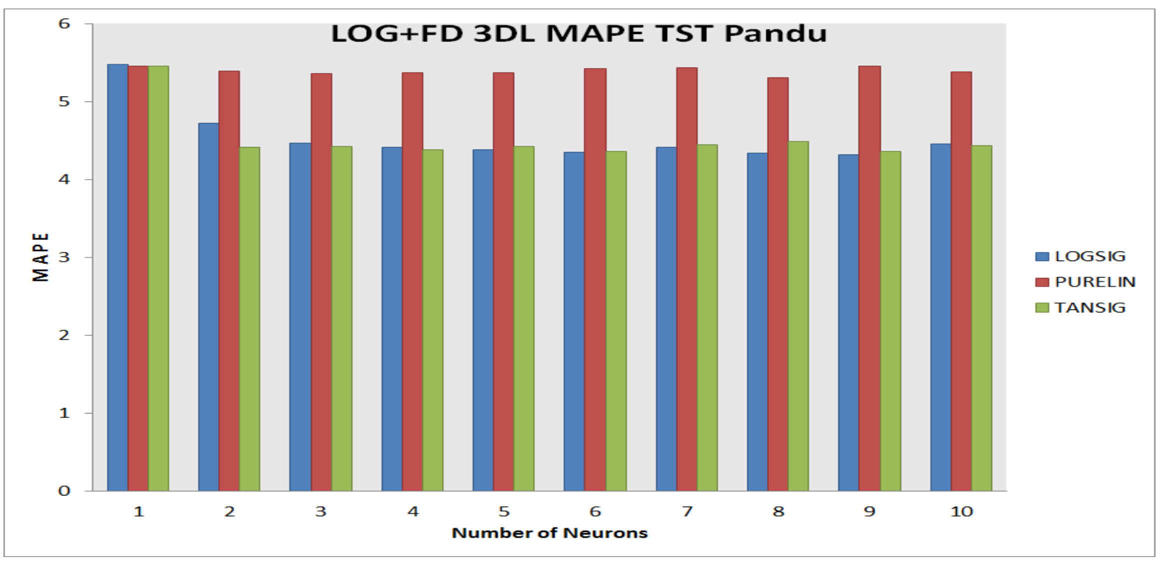


Fig. 4.36 Log + FD Data 3 day lag MAPE TST (Pandu)

4.5 Selection of Network – Dataset Combination

Thus in each type of datasets the best performance criteria are gathered together for the testing dataset as indicated in the Table no. 4.28 below. In Log Transformed Dataset, there are many networks very close to each other and the selection is not based only on lowest value of MAPE in the testing dataset, but its consistency is also taken into account.

Table 4.28 Comparison of Network Performance for Testing Dataset (Pandu)

Dataset	No. of Lagged Terms	Best Network Structure	LOGSIG		PURELIN		TANSIG	
			RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)
Raw	1	1-4-1	1765.99	4.87	2279.11	9.53	1817.12	5.02
	2	2-7-1	1115.43	3.39	18116.07	75.91	18488.26	81.57
	3	3-2-1	1117.38	3.39	1671.52	8.63	19227.07	82.66
Log Data	1	1-4-1	1020.61	2.67	1473.88	4.50	1018.50	2.76
	2	2-4-1	909.22	2.17	1383.70	4.14	946.19	2.23
	3	3-6-1	907.95	2.12	1379.97	4.09	913.66	2.21
Log + First Difference	1	1-8-1	1812.85	4.27	1956.57	5.44	1816.48	4.37
	2	2-4-1	1923.52	4.71	1957.12	5.43	1818.27	4.30
	3	3-8-1	1774.85	4.34	1951.85	5.30	1863.13	4.49

The values corresponding to training and validation dataset are collected and shown together in Table. No. 4.29.

Table 4.29 Comparison of Network Performance for Training Dataset (Pandu)

Dataset	No. of Lagged Terms	Best Network Structure	LOGSIG		PURELIN		TANSIG	
			RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)
Raw	1	1-4-1	1199.29	3.46	2121.47	19.08	1310.13	3.91
	2	2-7-1	1190.64	3.32	13101.48	55.59	13678.90	62.21
	3	3-2-1	1151.46	3.45	2053.04	19.15	18724.85	76.72
Log Data	1	1-4-1	1189.71	3.29	1582.56	5.12	1184.98	3.36
	2	2-4-1	1032.69	2.71	1467.76	4.75	996.86	2.68
	3	3-6-1	1002.80	2.67	1471.21	4.74	1031.30	2.73
Log + First Difference	1	1-8-1	2049.92	5.45	2168.63	6.51	2060.02	5.44
	2	2-4-1	2070.48	5.64	2169.25	6.53	2035.50	5.36
	3	3-8-1	1984.56	5.37	2174.34	6.51	2050.48	5.48

Thus through trial and error procedure by analyzing 540 trials, 270 for training and validation datasets and 270 for testing datasets, the **Log Transformed Dataset** and the **LOGSIG Network architecture** is found to be working in a very stable way than the Raw Dataset as well as the Log Transformed plus First Difference Dataset for the data at Pandu gauging station. The PURELIN architecture is found to give highly non-uniform results and shows high values of errors.

The final choice of network and data type is highlighted in both the training and testing datasets.Thus the final choice is :

Dataset : Log Transformed Dataset with Three Days Lag

Network Architecture : LOGSIG

Network Structure : 3 – 6 – 1

4.6 Testing of Selected Network

The consistency and robustness of the selected network is tested in three ways here.

1. Performance for high values
2. Performance for low values
3. Performance by interchanging Training and Testing Datasets.

High Values

Since the values of streamflow vary from 2432 m^3/s (minimum) to 61015 (maximum), the average being 17520 m^3/s, and noting that the values above 40000 m^3/s occur rarely, fixing >30000 m^3/s as the limit for high values, the filter is applied to all values together for the selected network and the RMSE and MAPE are computed.

Low Values

Fixing the limit for low values as < 5000 m^3/s, the filter is applied to all values predicted by the selected network and the RMSE and MAPE are computed. Following table shows the results.

Table 4.30 Testing for Consistency – High and Low Values

HIGH VALUES (>30000)		LOW VALUES (<5000)	
R M S E	M A P E	R M S E	M A P E
1739.13	3.08	140.40	2.07

Swapping The Training and Testing Datasets

Here the beginning 1/3rd data points, i.e. 1 to 2312 are taken as the testing dataset and end 2/3rd datapoints i.e. 2313 to 6936 are taken as the training dataset. The values for the next day, for datapoint 2313 to 6936 are taken as the target. All the datasets thus created are transformed to Logarithm of the value to the base 10. A new network of LOGSIG type with 6 neurons is created and trained with the training input and target. Then the validation is done with the training dataset without providing the target and the values predicted by the trained ANN are gathered as output for the validation. Testing dataset, also Log-Transformed is fed as input and results are obtained. The results of both the datasets are then compared with the actual values and RMSE and MAPE are computed. The statistical characteristics of the swapped datasets and the performance is shown in the next two tables.

Table 4.31 Statistical Characteristics of Swapped Datasets

Streamflow Value Q m^3/s	Training Dataset	Testing Dataset	All Dataset
Minimum	3008	2432	2432
Maximum	61015	51319	61015
Average	18545	16637	17904

Table 4.32 Results from Swapped Datasets

	RMSE TR	MAPE TR	RMSE TST	MAPE TST
After Swapping	1042.40	2.64	974.98	2.91
Before Swapping	1002.80	2.67	907.95	2.12

Here we see promising agreement between the results even when the datasets are interchanged validating the idea that if sufficiently large data with appropriate pre-processing technique is presented to a suitable ANN, that ANN can discern the datapattern and with the datapattern as only reference, can predict or forecast the future data with high accuracy, which may be difficult to achieve with the statistical or analytical models.

4.7 Comparison of the Predicted Streamflow with Actual Values

The following plot shows the entire data as forecast with the selected network in comparison with the actual data recorded at the guaging station Pandu. For better resolution, the plot is divided in four parts.

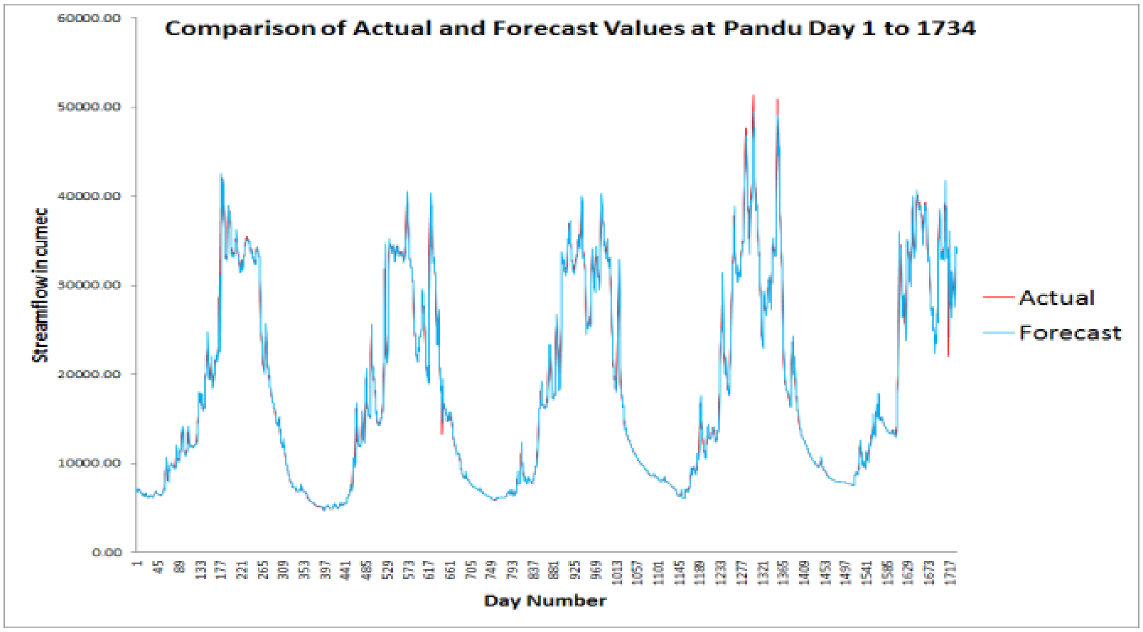


Fig. 4.37 Plot of Predicted and Actual Streamflow Day 1 – 1734 (Pandu)

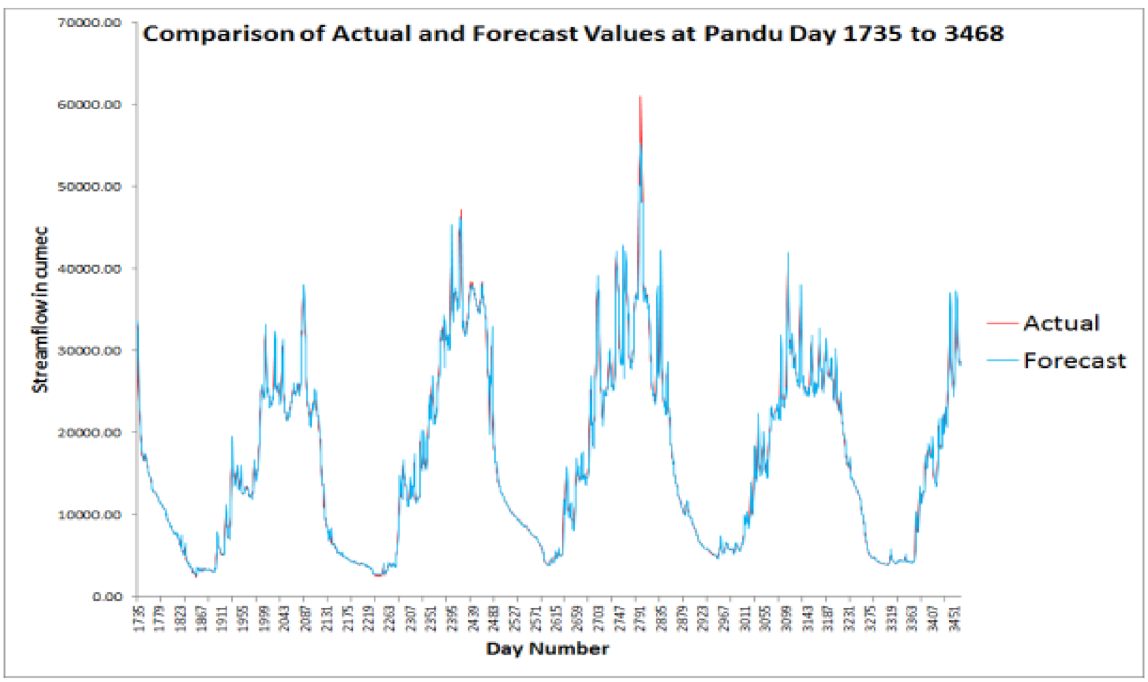


Fig. 4.38 Plot of Predicted and Actual Streamflow Day 1735 – 3468 (Pandu)

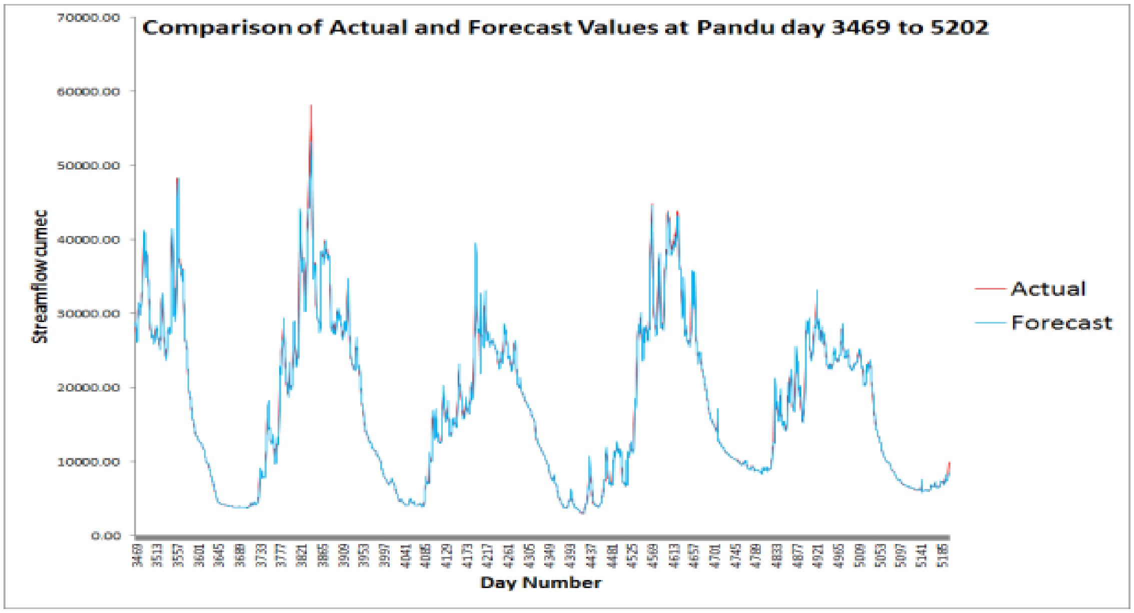


Fig. 4.39 Plot of Predicted and Actual Streamflow Day 3469 – 5202 (Pandu)

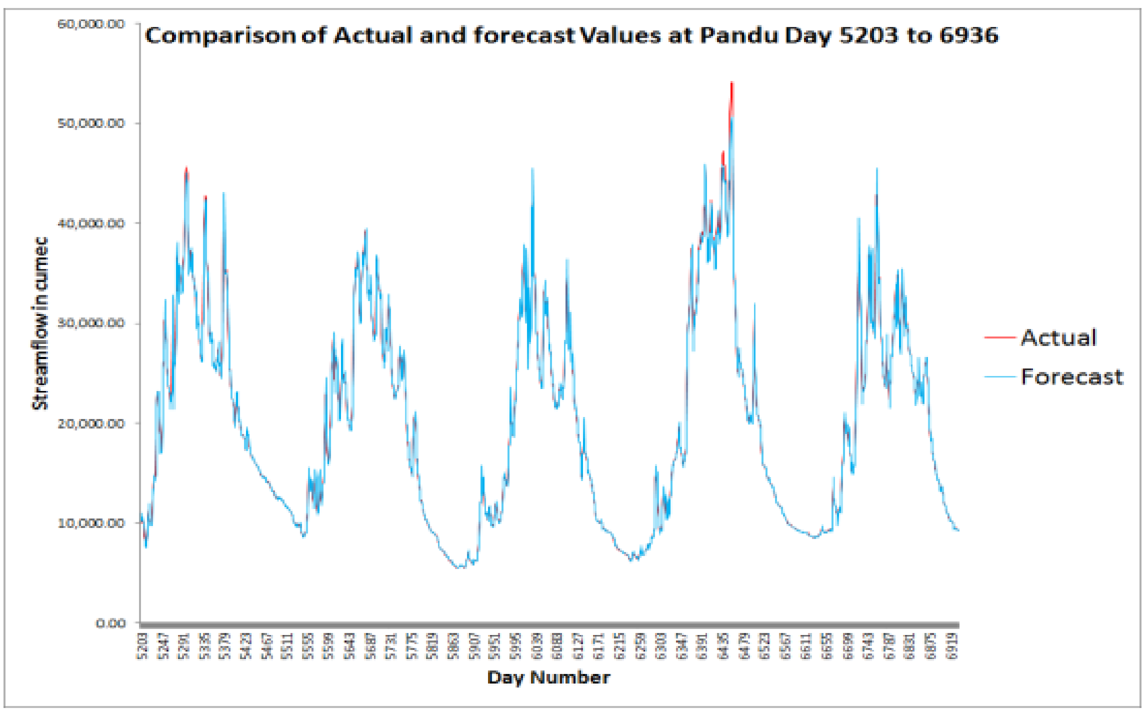


Fig. 4.40 Plot of Predicted and Actual Streamflow Day 5203 – 6936 (Pandu)

Enlarged view for better resolution is shown for parts where discrepancy is apparent.

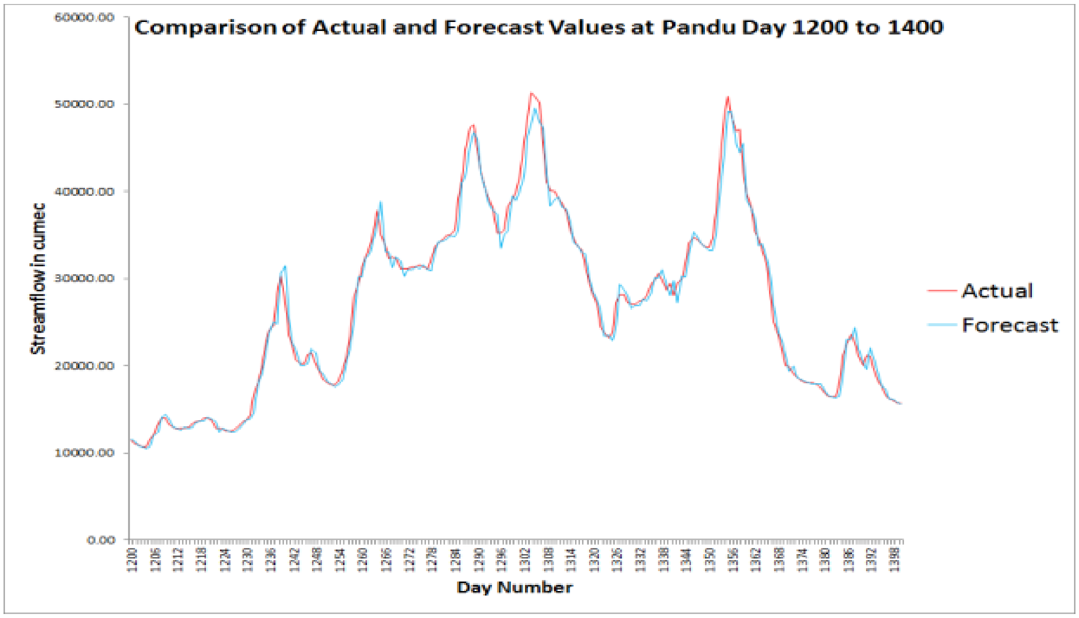


Fig. 4.41 Comparison of Actual and Predicted Streamflow Day 1200 – 1400 (Pandu)

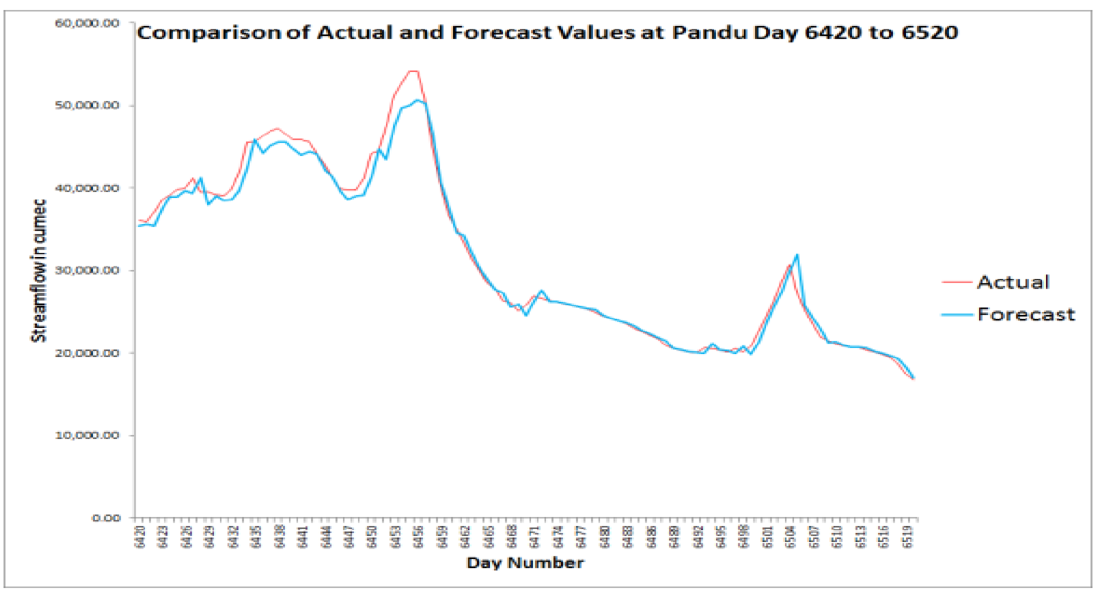


Fig. 4.42 Comparison of Actual and Predicted Streamflow Day 6420 – 6520 (Pandu)

In Fig. 4.41, it is seen near day no.s 1235-1240 that the forecast value is higher than the actual value. Whereas the two peaks between 1290-1320 show the forecast values lower than the actual values. Thus it can not be generalized whether the forecast is on the excessive side or not.

In Fig. 4.42 it is that the forecast is never drastically erroneous and in most of the flood peaks, the forecast is slightly lower, but on intermediate points it may be higher than the actual values. Hence for effective flood management this ANN can be usedas a forecasting tool very safely by applying a factor of safety between 1.2 to 1.5. This may entail flood relief operations a few days in advance, but as human life is of paramount importance, the extra effort and expense is definitely justified.

4.8 Results of Pancharatna Station

These are enumerated for the three types of datasets, viz. Raw, Log Transformed and Log plus First Difference.

4.8.1 Raw Data Sets - One Day Lag

Here the raw data of streamflow in m^3/s is given to the network as input and the network predicts the streamflow of the next day. The following table shows the evaluation results of the 30 trials resulting from the three architectures and by varying neurons in the hidden layer from 1 to 10.

It is observed here that PURELIN architecture performs poorly in comparison with LOGSIG and TANSIG, both of which perform almost equally, but LOGSIG can be said to perform only marginally better.

As already stated in the selection criteria in last chapter, basing on the performance of Testing dataset, the lowest values of MAPE and RMSE consistent with each other are shown in the shaded cells.

The results are graphically shown below for easy visualization.

Table 4.33 Raw Data 1 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2226.1	15.78	1803.43	13.69
2	1842.3	6.87	1493.23	6.01
3	19885.4	73.76	18845.7	76.21
4	1839.3	5.31	1533.75	4.62
5	1826.6	5.02	1480.45	4.38
6	16799.4	66.38	17052.0	70.56
7	17130.8	67.47	17814.8	72.27
8	10139.0	50.45	10791.6	54.64
9	19740.4	73.65	18804.8	76.18
10	19640.4	73.56	18769.1	76.15

Table 4.34 Raw Data 1 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	9979.9	92.27	9401.78	81.93
2	3179.8	34.29	2847.99	29.74
3	3180.0	34.30	2852.52	29.77
4	3180.7	34.05	2861.64	29.62
5	3179.8	34.25	2845.66	29.70
6	3187.3	35.35	2821.27	30.41
7	3182.0	33.82	2836.08	29.33
8	3180.3	34.51	2840.00	29.88
9	3180.7	34.28	2859.28	29.79
10	15297	67.47	15962.5	72.10

Table 4.35 Raw Data 1 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2228.47	16.51	1840.2	14.38
2	17027.9	58.49	15910	58.76
3	19111	90.43	8316.0	51.13
4	19885.4	73.76	18845.7	76.21
5	1813.87	5.15	1503.38	4.48
6	16150.6	45.14	15964.7	46.43
7	3542.71	5.31	1948.08	4.52
8	17710.8	71.65	17834.6	75.20
9	18760.6	51.59	17676.9	54.33
10	19652.9	73.57	18804.8	76.18

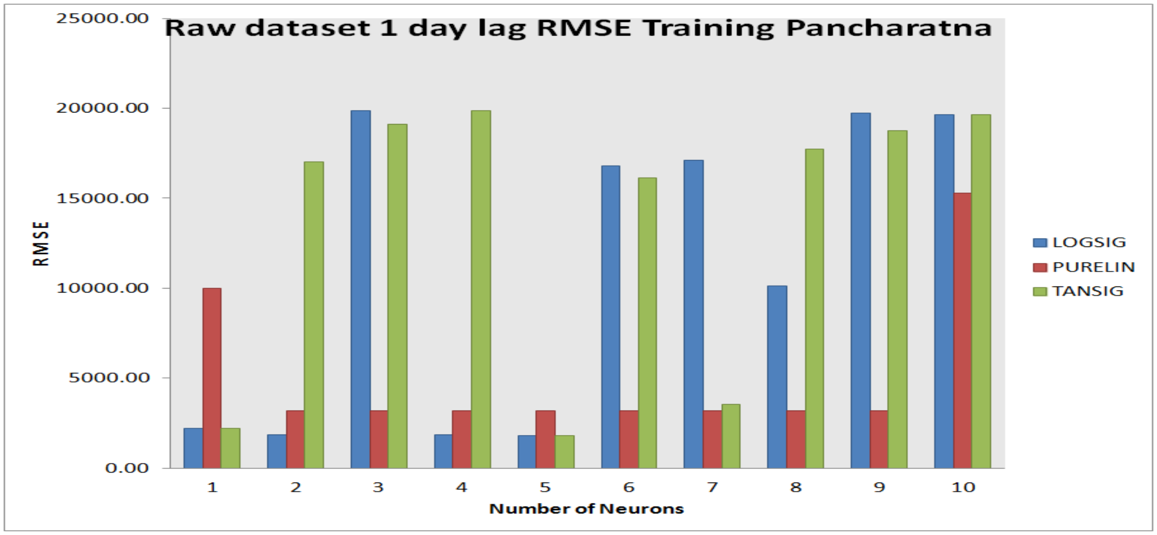


Fig. 4.43 Raw Data 1 day lag RMSE TR (Pancharatna)

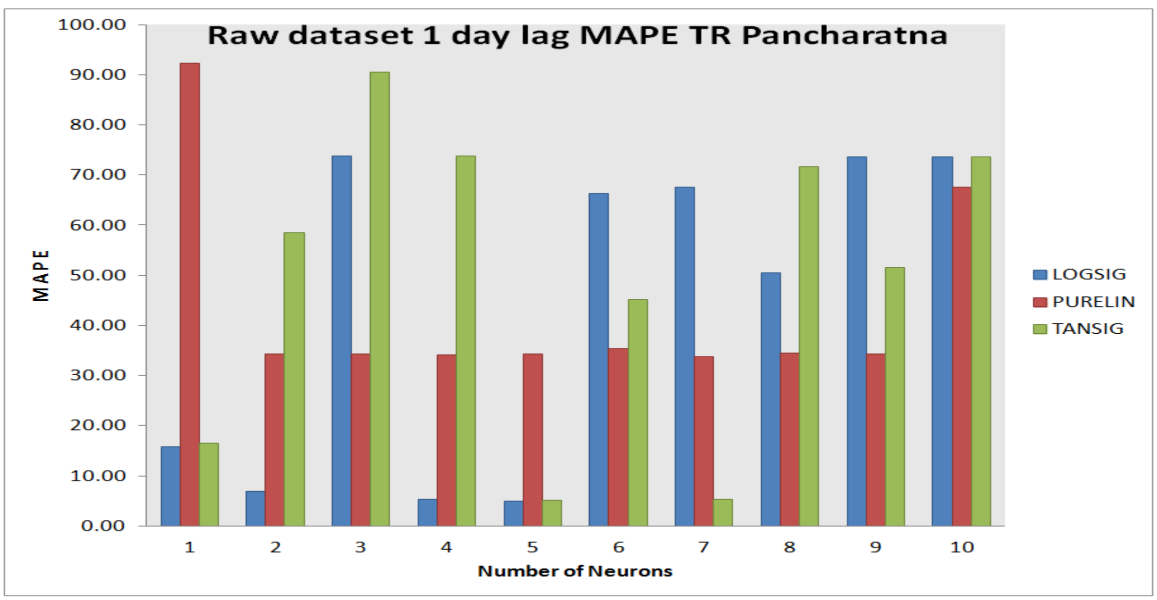


Fig. 4.44 Raw Data 1 day lag MAPE TR (Pancharatna)

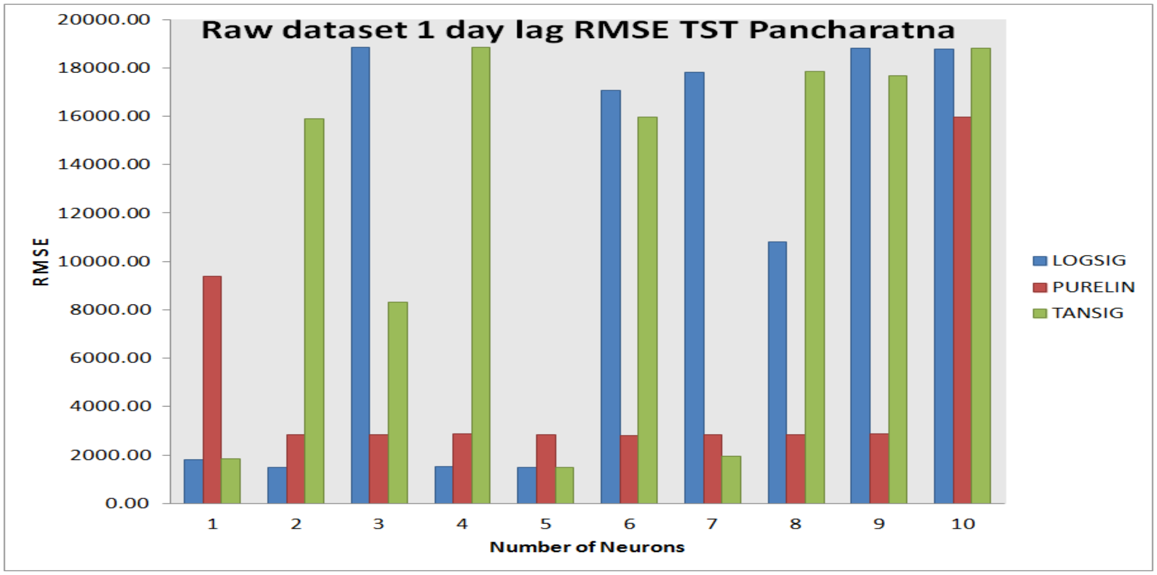


Fig. 4.45 Raw Data 1 day lag RMSE TST (Pancharatna)

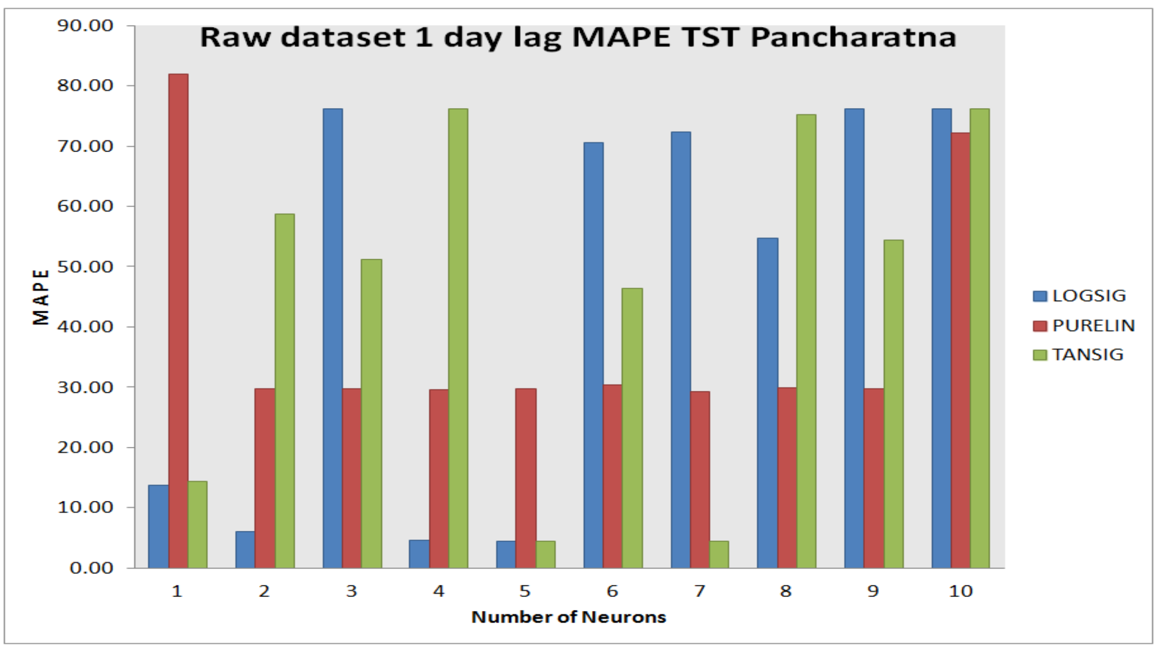


Fig. 4,46 Raw Data 1 day lag MAPE TST (Pancharatna)

4.8.2 Raw Data Sets - Two Day Lag

Here the streamflow of two consecutive days is given as the input and the ANN predicts the streamflow of the next day. Thus the ANN has access to more information about the time series. The results of the performance evaluation of the 30 networks formed with these data sets is shown in the tables below.

It is observed that LOGSIG architecture out performs both the PURELIN and TANSIG architectures. PURELIN architscture doesnot give the least error bot its performance is observed to be more consistent whereas some networks of the LOGSIG and TANSIG architectures give low errors, some give very high errors giving rise to inconsistent performance. The lowest values of errors in each architecture are shown by the shaded cells.

The results are graphically represented below for visualization.

Features similar to raw data sets as seen in the results of Pandu station can be observed here. Raw data sets are highly non linear and non stationary which is the limitation of ANNs in handling the datasets effectively.

Table 4.36 Raw Data 2 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2215.6	15.8	1747.2	13.6
2	18509	63.6	18434.7	67.5
3	19887	73.8	18842.3	76.2
4	1914.9	8.4	1508.4	7.2
5	19887	73.8	18842.3	76.2
6	1725.6	6.5	1400.6	5.5
7	1834.8	5.2	1446.8	4.3
8	19887	73.8	18842.3	76.2
9	1710.0	5.8	1369.8	4.8
10	1784.8	6.3	1418.9	5.4

Table 4.37 Raw Data 2 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	19960.00	97.90	19008.80	98.50
2	3178.00	34.60	2834.20	30.00
3	3178.70	33.80	2838.90	29.40
4	3183.10	34.70	2865.90	30.20
5	3177.70	34.00	2841.60	29.50
6	3194.50	36.70	2845.20	31.61
7	3178.50	34.40	2841.00	29.90
8	15384.00	54.70	15023.50	58.01
9	16904.00	65.80	16582.90	68.90
10	3179.40	34.10	2847.60	29.70

Table 4.38 Raw Data 2 day lag - TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2206.70	16.20	1790.70	14.10
2	13527.00	70.10	12039.00	89.90
3	1811.50	6.30	1459.30	5.40
4	1818.90	5.50	1462.00	4.70
5	19887.00	73.80	18842.00	76.20
6	20138.00	92.70	21541.00	97.70
7	22610.00	81.30	22526.00	84.30
8	19887.00	73.80	18842.00	76.20
9	19887.00	73.80	18842.00	76.20
10	19807.00	73.70	18801.00	76.20

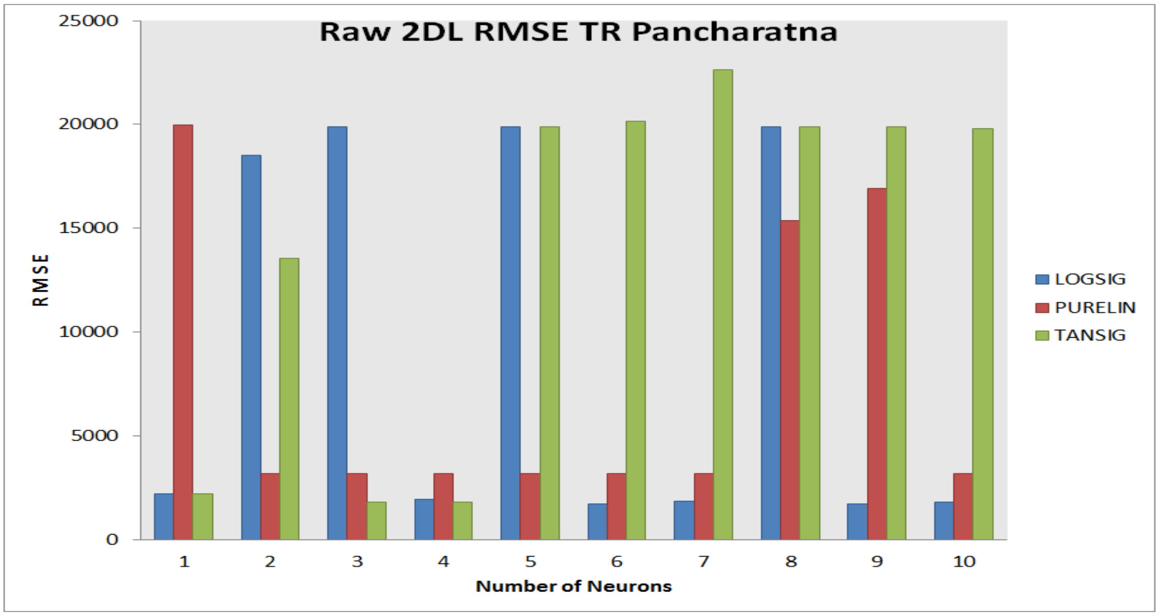


Fig. 4.47 Raw Data 2 day lag RMSE TR (Pancharatna)

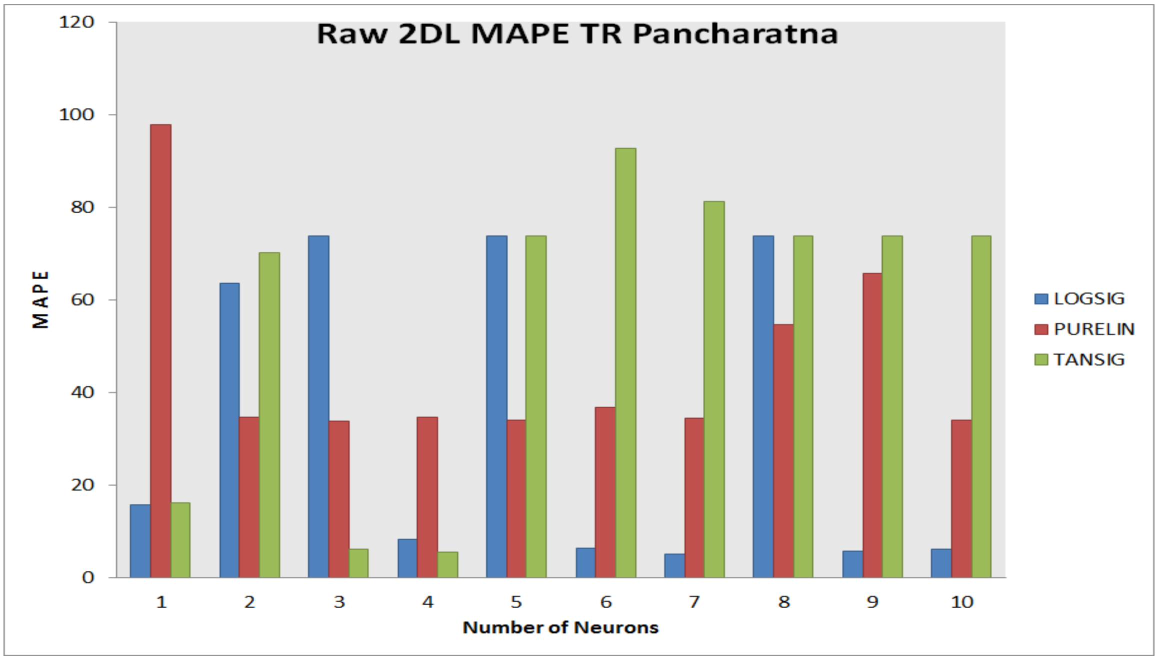


Fig. 4.48 Raw Data 2 day lag MAPE TR (Pancharatna)

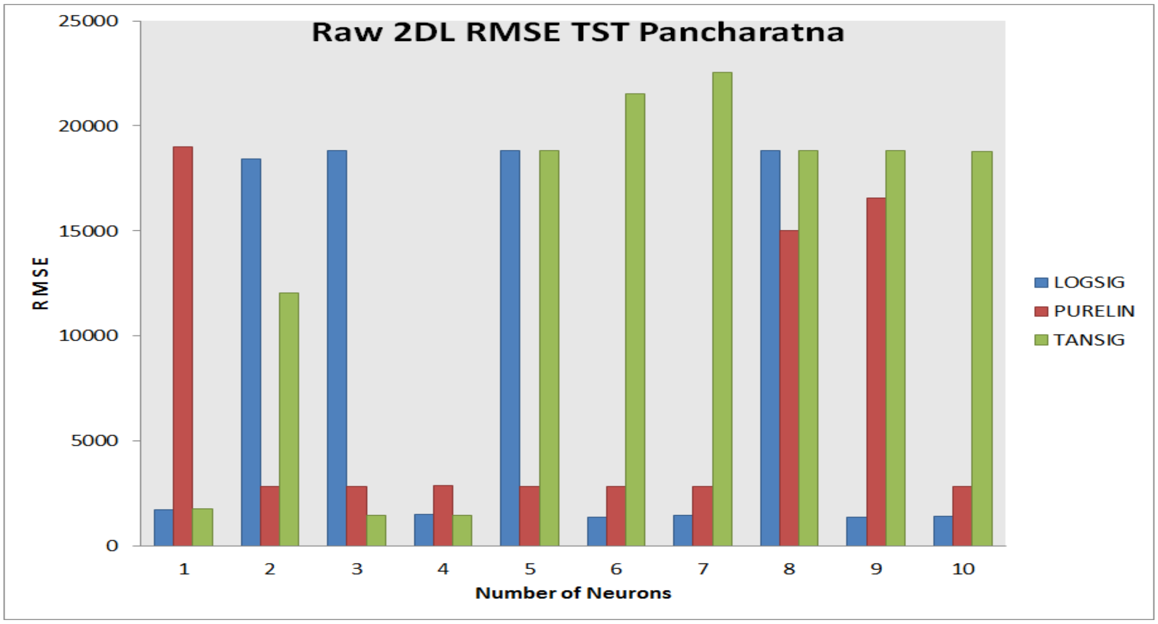


Fig. 4.49 Raw Data 2 day lag RMSE TST (Pancharatna)

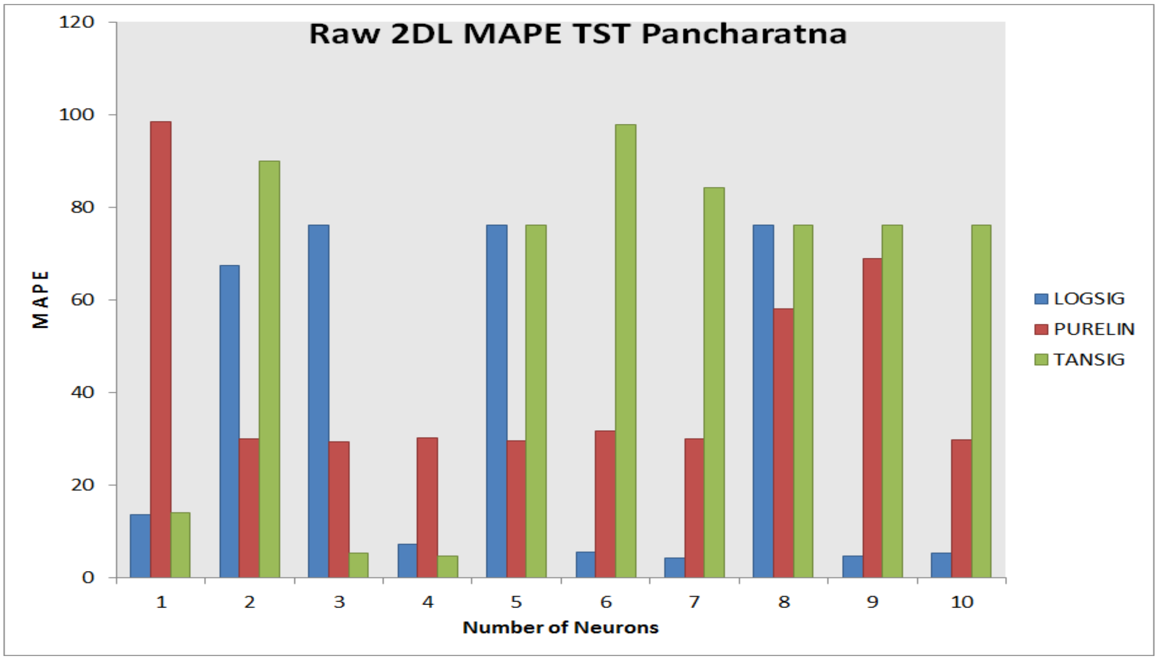


Fig. 4.50 Raw Data 2 day lag MAPE TST (Pancharatna)

4.8.3 Raw Data Sets – Three Day Lag

Here the streamflow of three consecutive days is entered into the ANN as input and the predicted value of the streamflow for the next, i.e. the fourth day is obtained as the output. The results of the different ANN- Dataset combinations are dhown in the table below.

Similar to the trend of previous datasets, here also non-consistent performance of LOGSIG and TANSIG architectures in comparison with PURELIN can be observed. Still, both of these architectures are able to give better prediction as indicated by very low values of bothy the RMSE and MAPE. Depending on the lowest MAPE values for the testing datasets consistent with RMSE and values of MAPE and RMSE for training and validation datasets are shown here by the cells with gray shading.

These results are also shown graphically.

Increase in the number of inputs causes better performance in the analysis at Pancharatna also. Thus we can safely generalise for the given time series at least, that it is desirable to have more number of lagged terms. This is true only up to a certain extent as in our work, in the pilot trials it was found that 3 inputs provides the optimum and data overload may lead to over learning and over fitting and the problems associated with it.

Table 4.39 Raw Data 3 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2194.70	15.50	1748.60	13.50
2	2095.40	13.80	1647.60	11.90
3	6048.20	69.60	5142.70	58.70
4	1811.70	7.20	1449.30	6.10
5	19896.10	73.80	18798.40	76.20
6	1804.10	6.00	1383.50	5.00
7	19888.50	73.80	18839.40	76.20
8	17004.10	66.60	17233.30	70.80
9	1969.50	6.10	1466.80	5.20
10	1701.70	4.80	1310.20	3.90

Table 4.40 Raw Data 3 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3179.80	34.40	2829.80	29.80
2	3176.70	34.30	2842.50	29.80
3	3177.60	34.10	2847.20	29.70
4	3177.90	34.50	2842.20	29.90
5	3176.80	34.40	2837.50	29.90
6	3178.40	34.00	2823.10	29.50
7	17120.20	70.50	15796.10	67.50
8	16135.70	68.90	16641.10	73.10
9	3178.30	33.60	2838.40	29.30
10	3178.10	34.00	2831.10	29.60

Table 4.41 Raw Data 3 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	10672.80	97.40	9630.50	88.00
2	1796.80	6.30	1419.20	5.40
3	19888.50	73.80	18839.40	76.20
4	18993.20	73.00	18545.20	75.90
5	15256.30	97.90	14256.70	87.40
6	17732.30	71.70	17882.60	75.20
7	1689.80	5.70	1372.20	4.90
8	1654.50	5.40	1342.00	4.50
9	18569.00	73.00	18384.10	76.00
10	1683.70	4.70	1365.20	3.80

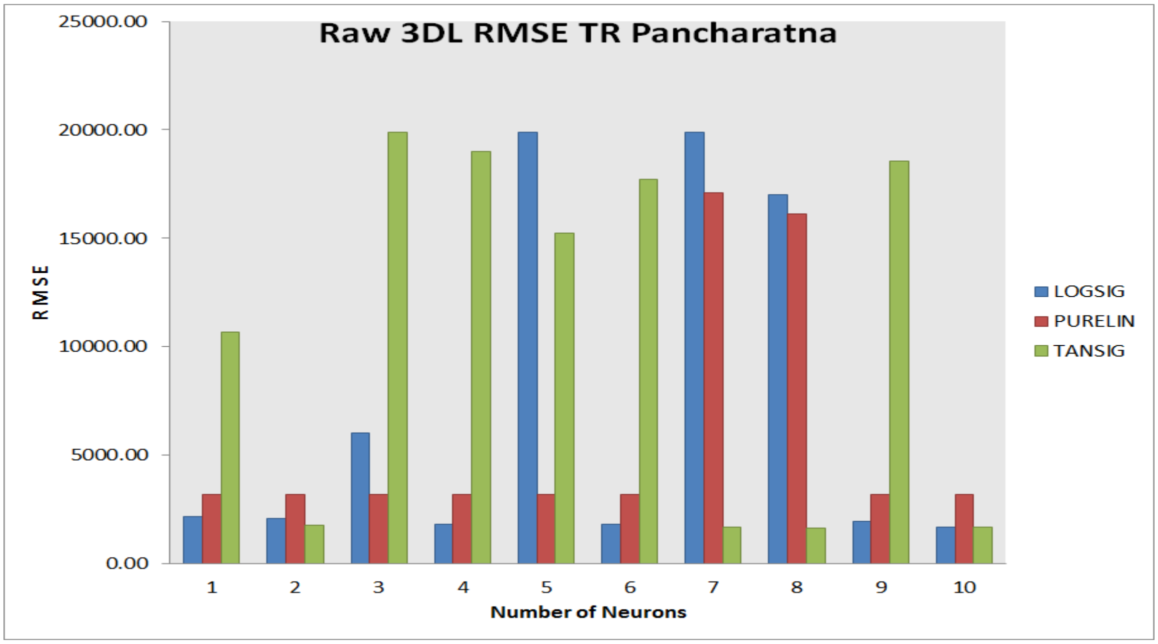


Fig. 4.51 Raw Data 3 day lag RMSE TR (Pancharatna)

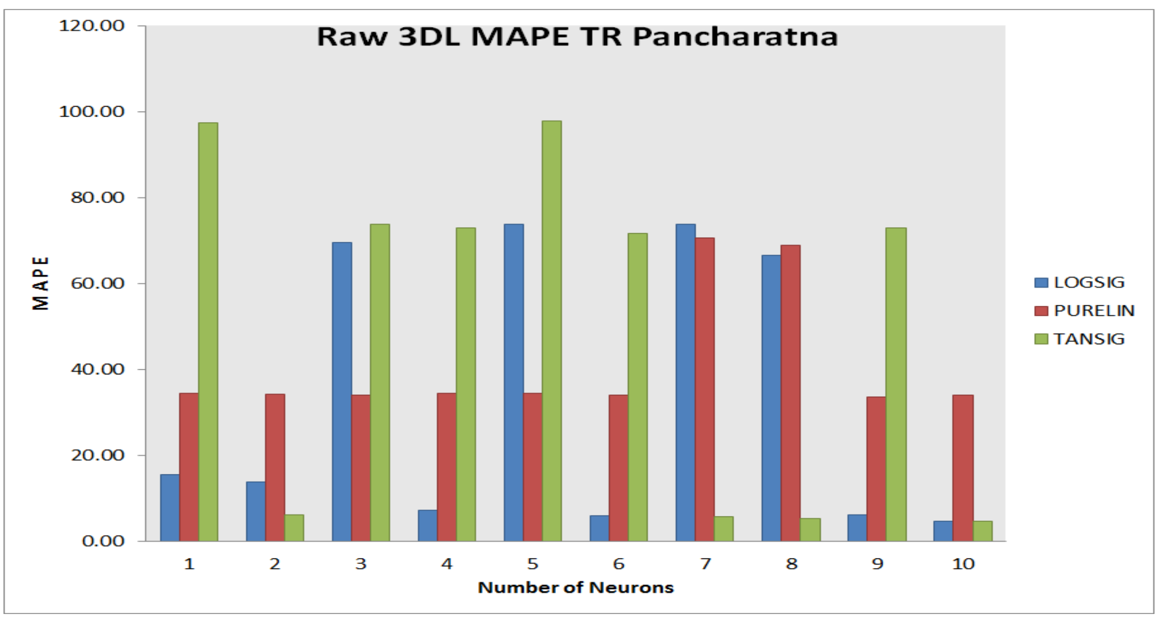


Fig. 4.52 Raw Data 3 day lag MAPE TR (Pancharatna)

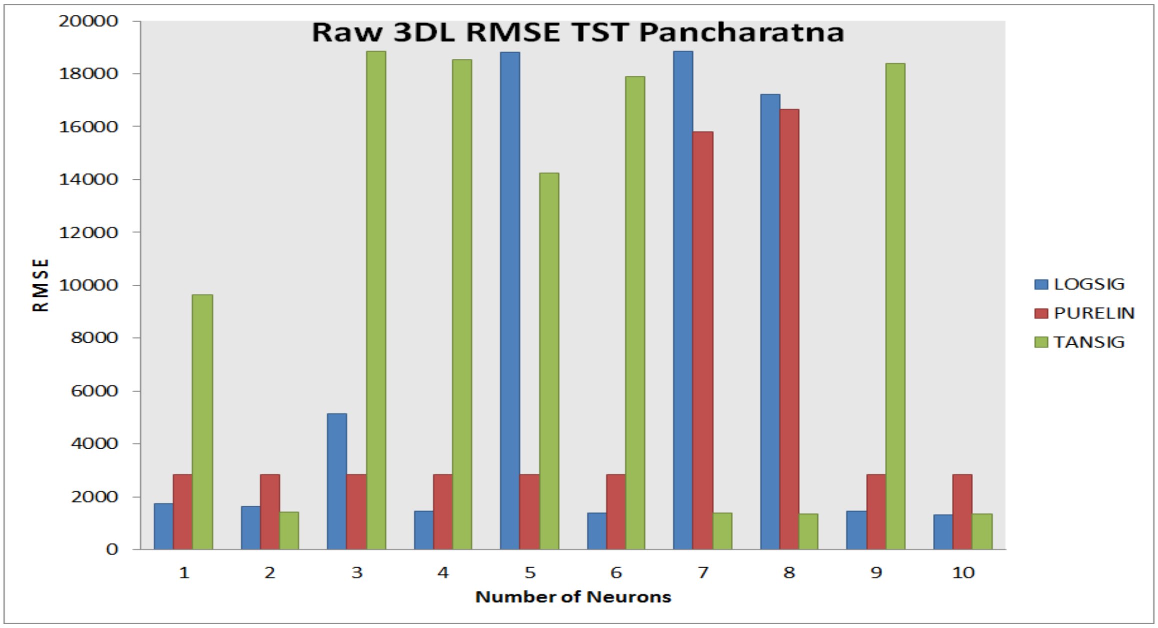


Fig.4.53 Raw Data 3 day lag RMSE TST (Pancharatna)

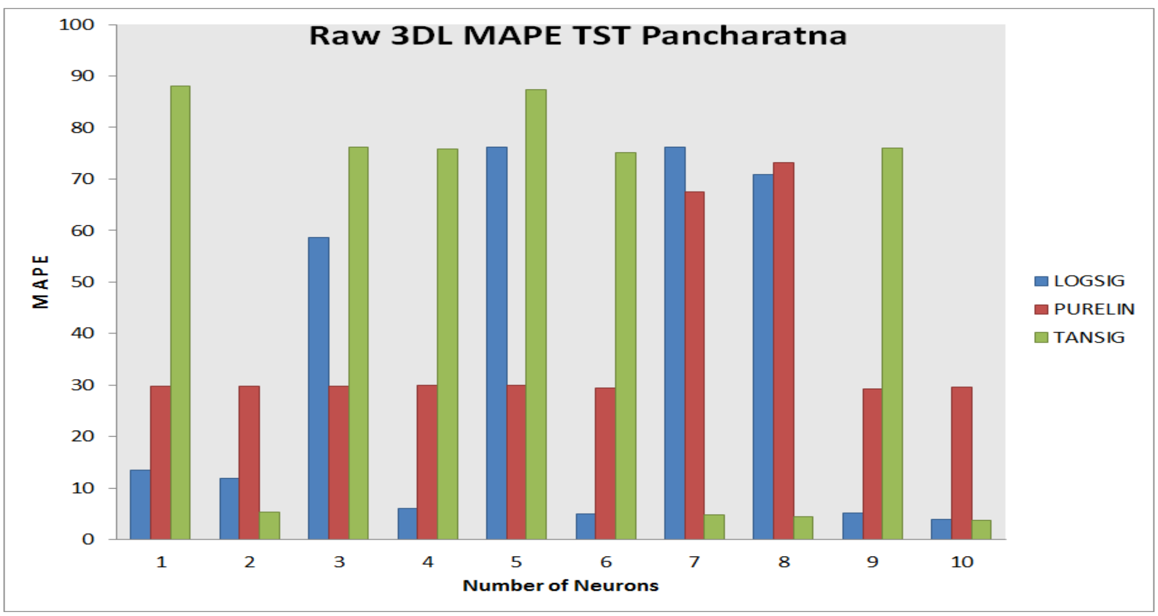


Fig. 4.54 Raw Data 3 day lag MAPE TST (Pancharatna)

4.8.4 Log Transformed Data – One Day Lag

Here the logarithm of each data point is taken to the base 10 and the data are transformed by this pre-processing technique. Since the range of the data variation is very large in this particular situation of the Himalayan river, and the skewness coefficient is also high, logarithmic data is a logical choice of pre-processing technique as it will flatten the data spread into a much thinner band thus it may enable the ANN to perform better.

The table below shows the results.

Substantially improved and consistent performance of the ANNs compared to that of raw dataset is seen in this analysis. It is also seen that LOGSIG and TANSIG perform almost equally well compared to PURELIN architecture, which does not perform so well. The lowest errors are given by LOGSIG network and the network in each category with minimum error is indicated by highlighting.

The results are also visually represented.

Features similar to Pandu can be observed here as the randomness of performance which is seen from plots in Fig. 4.43 onwards up to Fig. 4.54 suddenly vanishes and we can see more robust and reliable performance from the plots in Fig. 4.55 onwards. Since robustness is also important rather than only accuracy of prediction taken in isolation, the log transformed datasets give definitely a better overall picture than the raw datasets in both performance as well as in the robustness and reliability of the ANNs in dealing with previously unseen data.

Table 4.42 Log Data 1 day lag LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2398.90	6.64	1825.05	5.96
2	1855.00	4.80	1460.72	4.13
3	1842.80	4.73	1460.08	4.06
4	1837.90	4.85	1473.95	4.25
5	1826.80	4.79	1467.93	4.17
6	1831.00	4.80	1476.73	4.18
7	1824.60	4.81	1466.26	4.22
8	1825.40	4.72	1471.09	4.08
9	1830.10	4.76	1492.87	4.23
10	1822.90	4.83	1494.95	4.26

Table 4.43 Log Data 1 day lag –PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2280.90	6.72	1732.33	5.94
2	2277.50	6.74	1737.91	5.99
3	2274.70	6.74	1736.30	5.98
4	2274.00	6.74	1734.56	5.97
5	2272.20	6.75	1736.35	5.99
6	2273.60	6.74	1735.79	5.98
7	2278.10	6.73	1737.21	5.98
8	2272.30	6.75	1739.43	6.01
9	2281.00	6.73	1736.20	5.96
10	2275.90	6.73	1734.34	5.97

Table 4.44 Log Data 1 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2398.80	6.61	1818.30	5.91
2	2117.40	6.41	1620.10	5.68
3	1988.10	6.70	1626.10	6.00
4	1822.80	4.77	1470.70	4.13
5	1821.70	4.75	1469.00	4.14
6	1845.30	4.78	1464.50	4.12
7	1819.20	4.78	1475.70	4.16
8	1822.10	4.76	1468.70	4.13
9	1820.80	4.78	1816.70	4.73
10	1821.10	4.75	1879.30	4.78

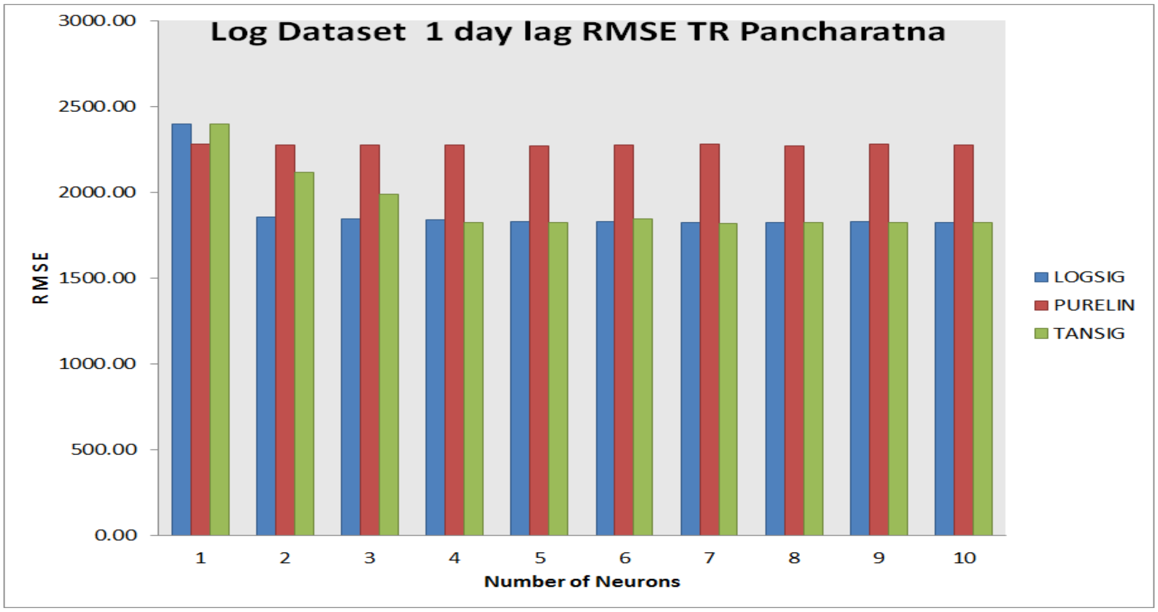


Fig. 4.55 Log Data 1 day lag RMSE TR (Pancharatna)

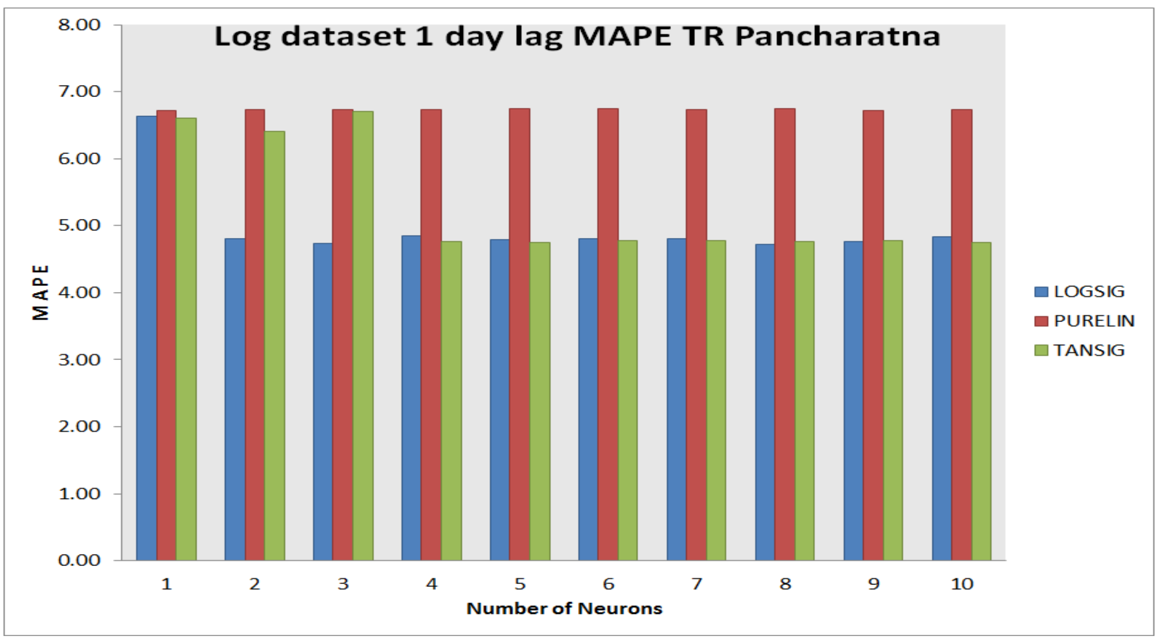


Fig. 4.56 Log Data 1 day lag MAPE TR (Pancharatna)

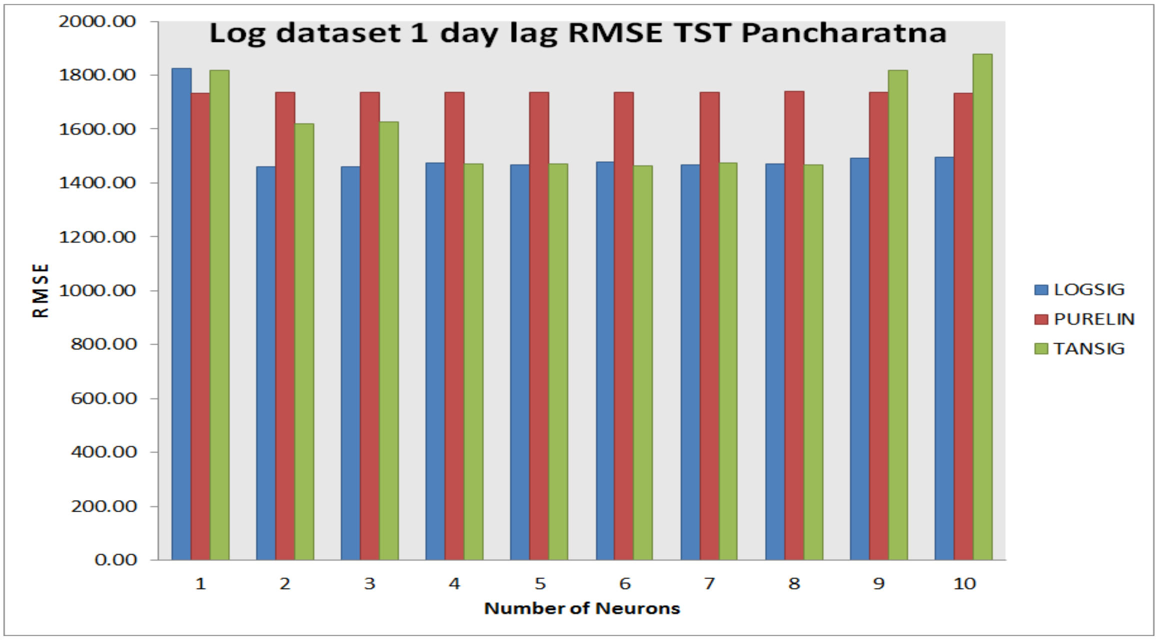


Fig. 4.57 Log Data 1 day lag RMSE TST (Pancharatna)

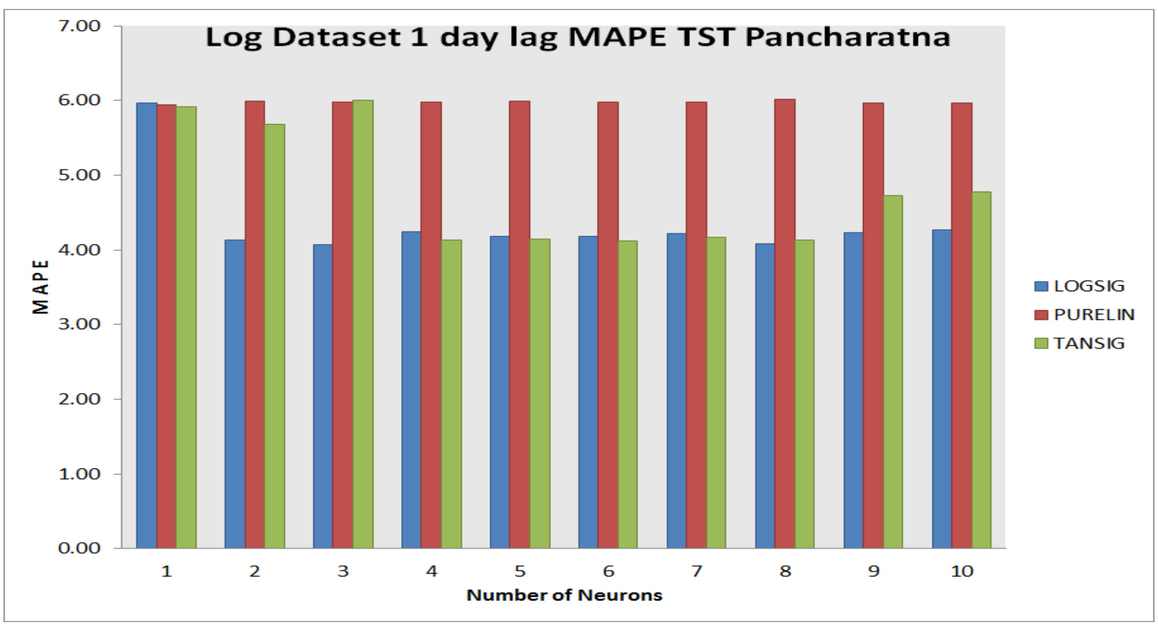


Fig. 4.58 Log Data 1 day lag MAPE TST (Pancharatna)

4.8.5 Log Transformed Data – Two Day Lag

Here the log-transformed data of two consecutive days is given as input and the next day's streamflow is predicted. It is again re-transformed in the original units and then comapared with the original value. The results are represented in the observation table and by graphics following the observation table. The gray shading in the table shows the networks from each architecture giving optimum performance for this dataset according to the performance criteria adopted

Here the trend is similar but an even lower value of error is obtained. In this case the lowest error is given by the TANSIG network. The network with minimum error in each category is highlighted.

These results are represented graphically below.

Table 4.45 Log Data 2 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2366.40	6.51	1782.50	5.85
2	1893.50	4.73	1433.20	3.99
3	1807.60	4.66	1423.76	3.99
4	1812.10	4.65	1417.36	3.97
5	1870.60	4.69	1430.49	3.90
6	1775.20	4.52	1425.71	3.67
7	1845.30	4.57	1394.37	3.77
8	1831.40	4.82	1432.34	4.10
9	1890.40	4.73	1440.73	3.94
10	1745.70	4.49	1411.45	3.72

Table 4.46 Log Data 2 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2237.00	6.63	1692.13	5.85
2	2244.60	6.61	1700.77	5.89
3	2249.50	6.63	1700.77	5.89
4	2241.70	6.60	1687.40	5.84
5	2257.50	6.60	1696.91	5.84
6	2247.00	6.61	1692.81	5.85
7	2243.00	6.63	1698.78	5.90
8	2240.10	6.65	1700.91	5.93
9	2241.60	6.63	1695.79	5.89
10	2248.30	6.62	1697.77	5.87

Table 4.47 Log Data 2 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2367.60	6.53	1787.20	5.87
2	2360.90	6.51	1778.50	5.84
3	1835.00	4.67	1416.80	3.98
4	1806.30	4.56	1428.40	3.89
5	1811.50	4.63	1439.60	3.98
6	1802.20	4.58	1403.50	3.84
7	1756.30	4.30	1725.70	3.59
8	1765.70	4.51	1404.10	3.69
9	1750.00	4.43	1427.90	3.85
10	1732.20	4.36	1391.40	3.52

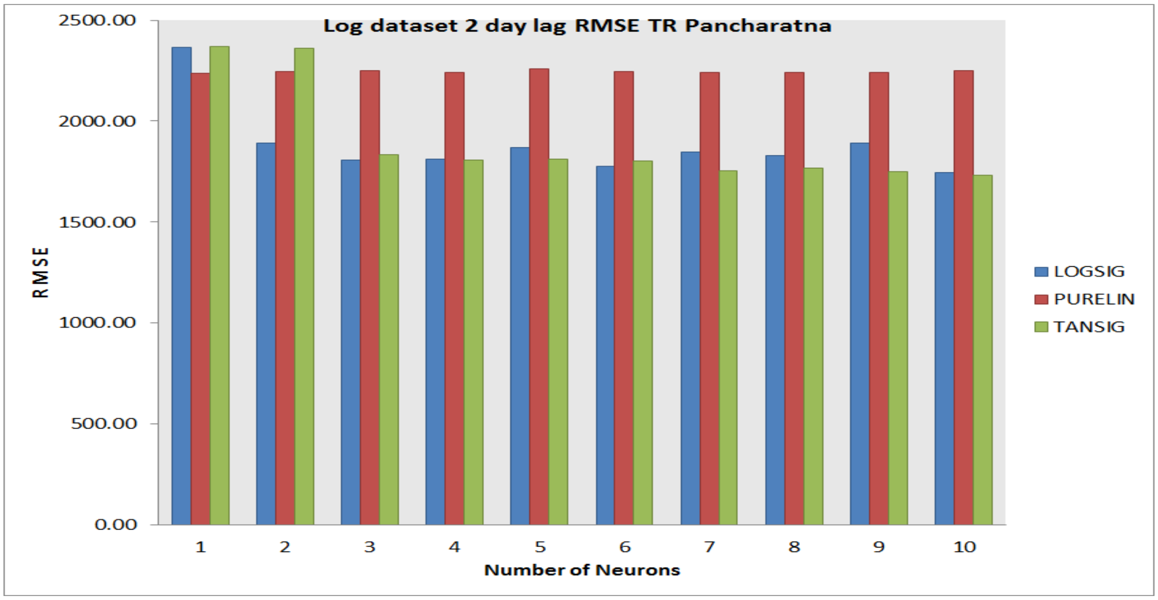


Fig. 4.59 Log Data 2 day lag RMSE TR (Pancharatna)

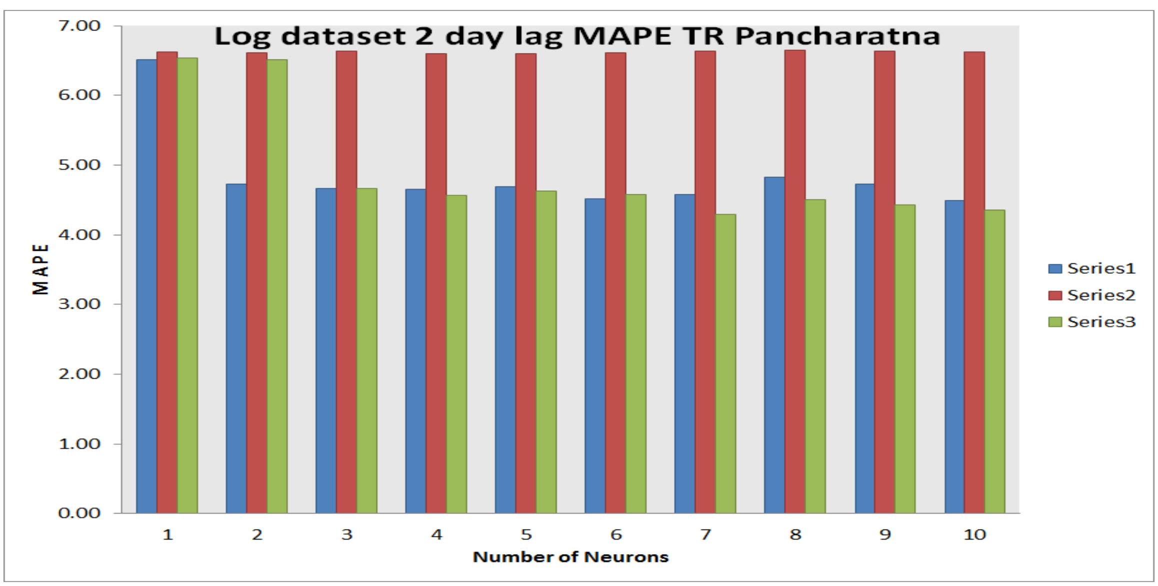


Fig. 4.60 Log Data 2 day lag MAPE TR (Pancharatna)

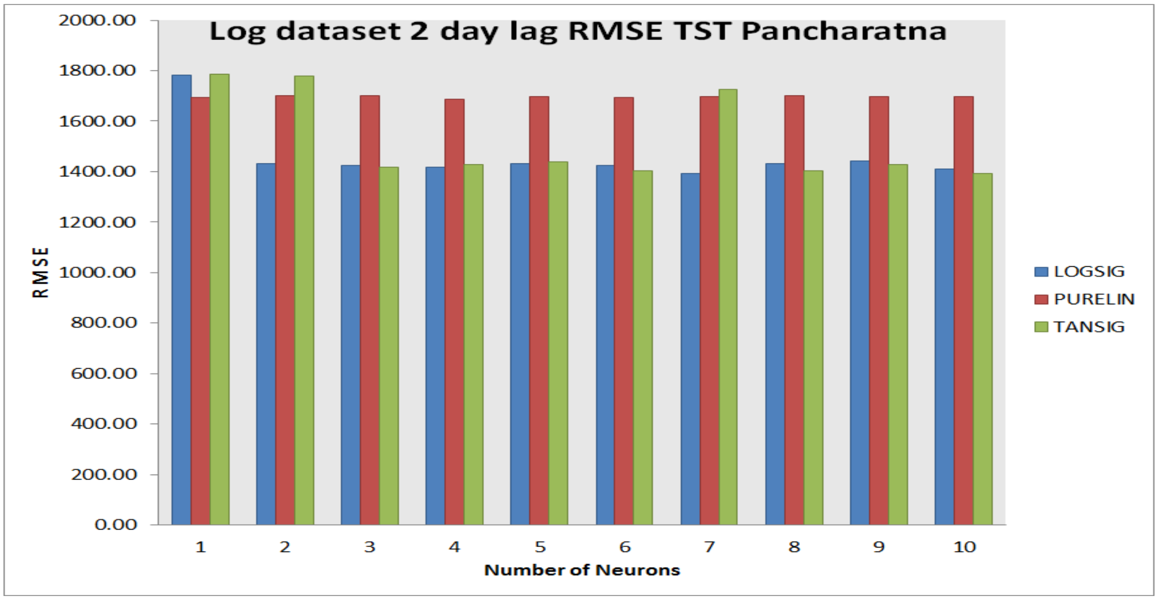


Fig. 4.61 Log Data 2 day lag RMSE TST (Pancharatna)

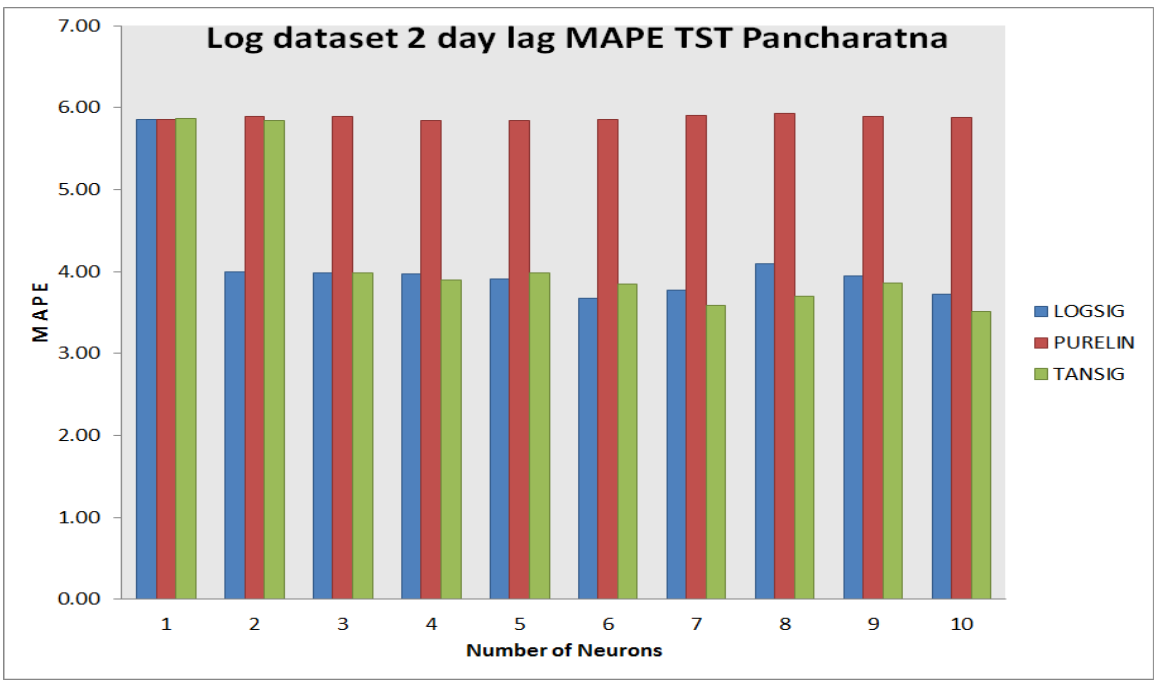


Fig. 4.62 Log Data 2 day lag MAPE TST (Pancharatna)

4.8.6 Log Transformed Data – Three Day Lag

Here the streamflow values are transformed in log to the base 10 and dataset for input contains data for three consecutive days, the value for the next day being predicted by each network. These values are re- transformed into original units and then compared with the observed values for calculating RMSE and MAPE for both Training-Validation as well as Testing datasets. The results are shown in the table below.

It is seen that LOGSIG gives the lowest error value and PURELIN performance is poor compared to both LOGSIG and TANSIG. Due to three day lag, the error is further reduced by giving three input values. The best performing network results are marked by highlighting. Here LOGSIG network gives the least error.

These results are also shown graphically.

Table 4.48 Log Data 3 day lag - LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2346.60	6.46	1762.40	5.79
2	1816.60	4.54	1387.36	3.77
3	1919.70	4.61	1701.79	4.02
4	1844.80	4.63	1411.31	3.90
5	1717.80	4.30	1337.90	3.46
6	2107.50	4.65	1390.19	3.86
7	1812.40	4.46	1428.68	3.56
8	1809.30	4.58	1395.33	3.87
9	1794.80	4.52	1388.77	3.79
10	1795.20	4.55	1412.31	3.81

Table 4.49 Log Data 3 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2227.70	6.64	1680.32	5.88
2	2220.20	6.61	1677.36	5.87
3	2227.60	6.59	1676.35	5.83
4	2227.20	6.60	1675.70	5.84
5	2211.00	6.61	1673.17	5.88
6	2214.20	6.61	1671.84	5.87
7	2223.40	6.60	1674.62	5.84
8	2211.50	6.61	1670.87	5.87
9	2213.80	6.61	1667.74	5.85
10	2223.90	6.60	1674.38	5.84

Table 4.50 Log Data 3 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	2370.40	6.45	1776.10	5.77
2	1837.60	4.66	1388.60	3.94
3	1831.80	4.58	1385.80	3.80
4	1861.90	4.74	1397.90	3.96
5	1824.80	4.55	1401.00	3.73
6	1791.50	4.54	1382.70	3.80
7	1792.80	4.62	1392.50	3.87
8	1798.90	4.53	1375.10	3.68
9	2026.80	5.19	1490.80	4.31
10	1719.80	4.33	1356.70	3.58

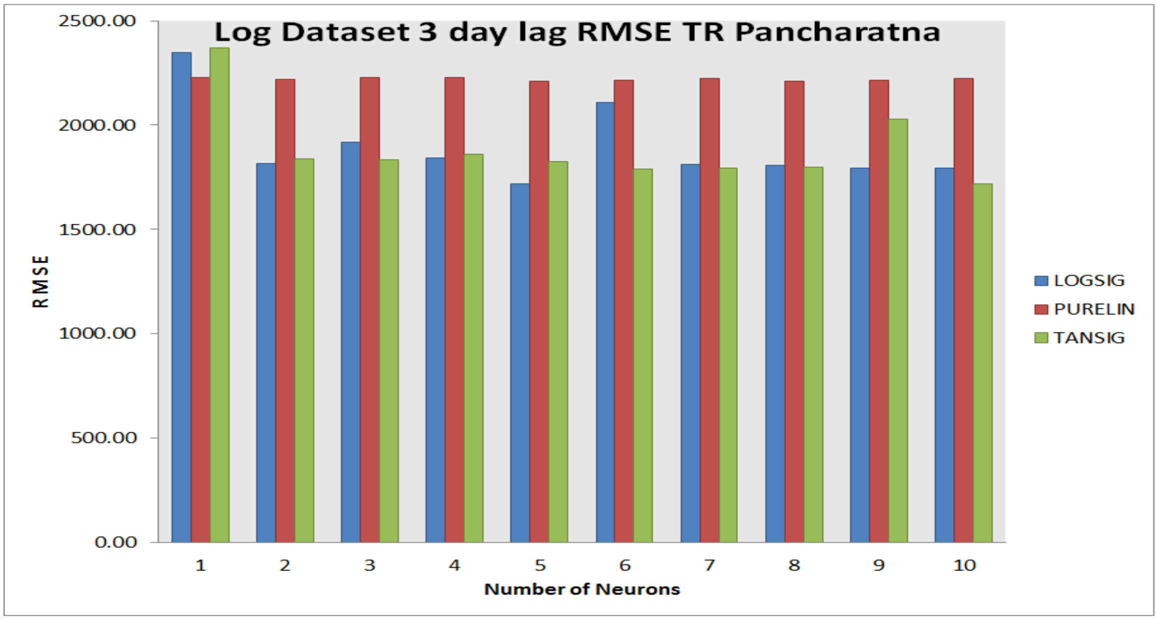


Fig. 4.63 Log Data 3 day lag RMSE TR (Pancharatna)

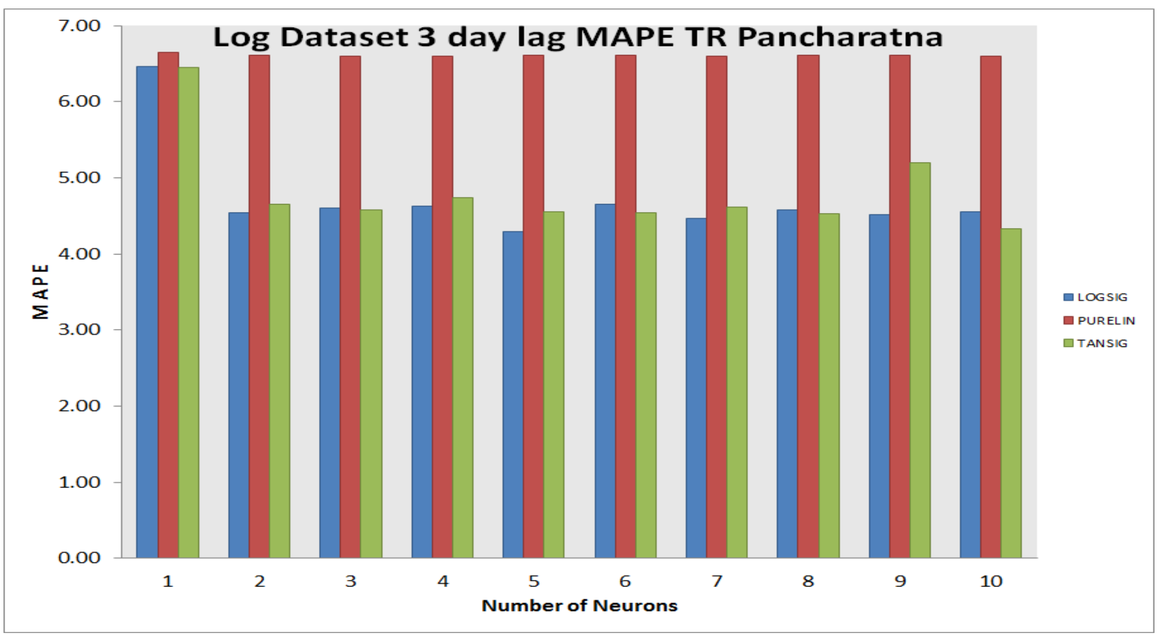


Fig. 4.64 Log Data 3 day lag MAPE TR (Pancharatna)

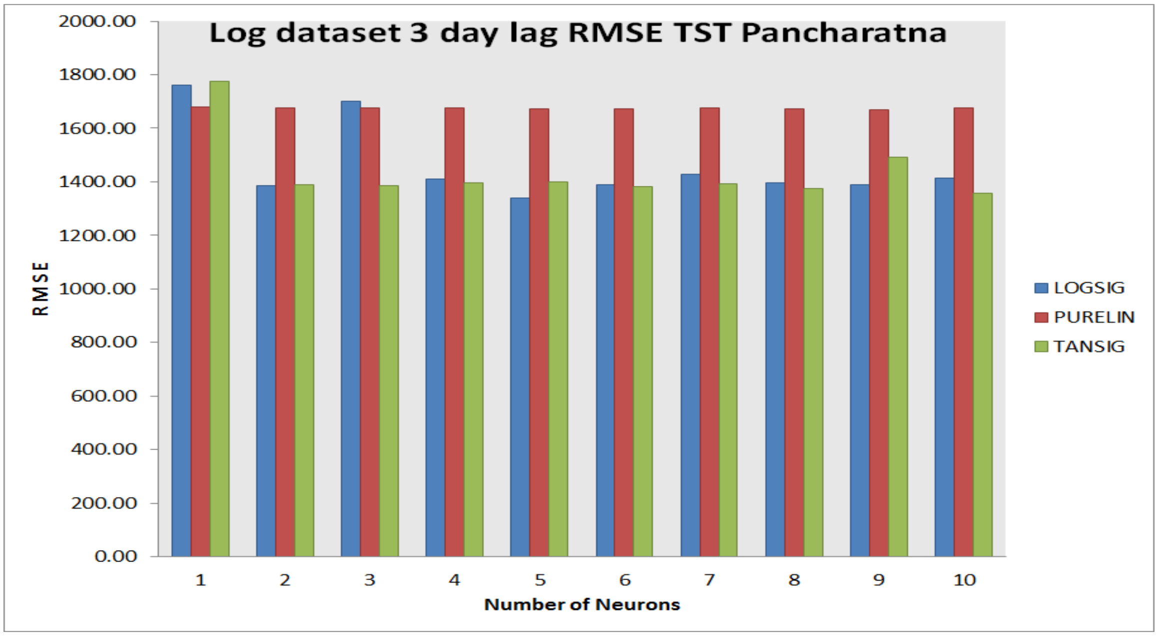
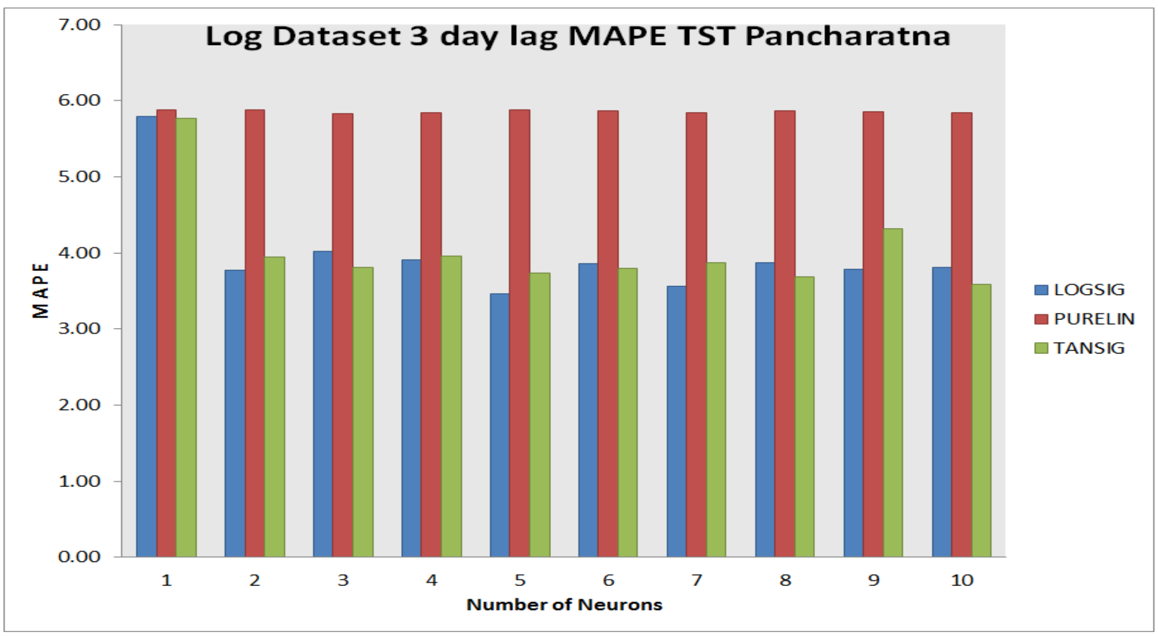


Fig. 4.65 Log Data 3 day lag RMSE TST (Pancharatna)



.Fig. 4.66 Log Data 3 day lag MAPE TST (Pancharatna)

4.8.7 Log plus First Difference – One Day Lag

Here the first difference, i.e. (x_i – x_{i-1}) is calculated after taking logarithm of each streamflow value and this first difference is added to the x_i value. For the very first value, i.e. for x_1, the first difference is taken as zero. Then this data are arranged in 1-day, 2-day and 3-day lag pattern and appropriate 2/3rd and 1/3rd data points are separated as training/validation set and testing set respectively.

After giving the one-day lag input, the results from various networks are re-transformed into corresponding original form and then compared for analysis and computations of error criteria.

The tables below show the result of this category.

Here the results show greater errors than the Log Transformed Dataset. Still, the results or the performance is not unstable and a stable performance for all the three activating functions can be observed. Here LOGSIG and TANSIG perform almost equally well and only marginally better than PURELIN.

The minimum error is highlighted in each category.

The results are also graphically represented.

Table 4.51 Log + FD Data 1 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3831.10	9.96	3073.57	8.38
2	3819.50	9.96	3074.59	8.38
3	3815.70	10.00	3069.19	8.44
4	3773.90	9.34	3018.92	7.56
5	3739.90	9.35	3010.55	7.56
6	3819.90	9.39	3105.61	7.59
7	3922.70	9.61	3133.49	7.81
8	3812.50	9.44	3175.03	8.06
9	3731.10	9.39	3097.60	7.67
10	3740.60	9.42	3089.36	7.76

Table 4.52 Log + FD Data 1 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3826.90	9.95	3073.50	8.37
2	3828.40	9.95	3072.00	8.37
3	3830.80	9.95	3069.70	8.35
4	3829.80	10.03	3090.70	8.52
5	3827.40	9.98	3079.20	8.43
6	3828.80	9.95	3071.50	8.37
7	3830.10	9.99	3074.80	8.43
8	3826.80	9.94	3072.50	8.36
9	3826.80	9.95	3074.30	8.38
10	3829.50	9.92	3068.30	8.32

Table 4.53 Log + FD Data 1 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3849.10	10.05	3081.70	8.49
2	3820.50	10.00	3072.50	8.44
3	3808.50	9.43	3090.30	7.69
4	3770.60	9.44	2989.70	7.62
5	3814.30	10.09	3073.70	8.48
6	3765.20	9.45	3008.50	7.63
7	3914.70	10.06	3113.70	8.40
8	4071.30	10.59	3719.40	8.78
9	3782.80	9.53	3188.90	7.82
10	3829.30	9.39	3149.50	7.84

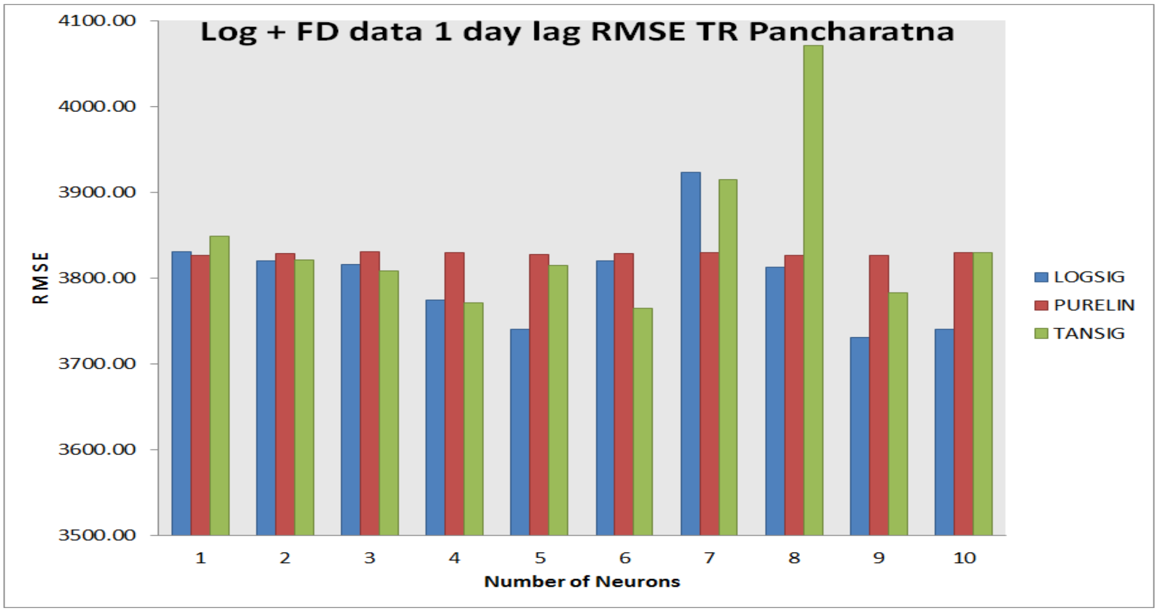


Fig.4.67 Log + FD Data 1 day lag RMSE TR (Pancharatna)

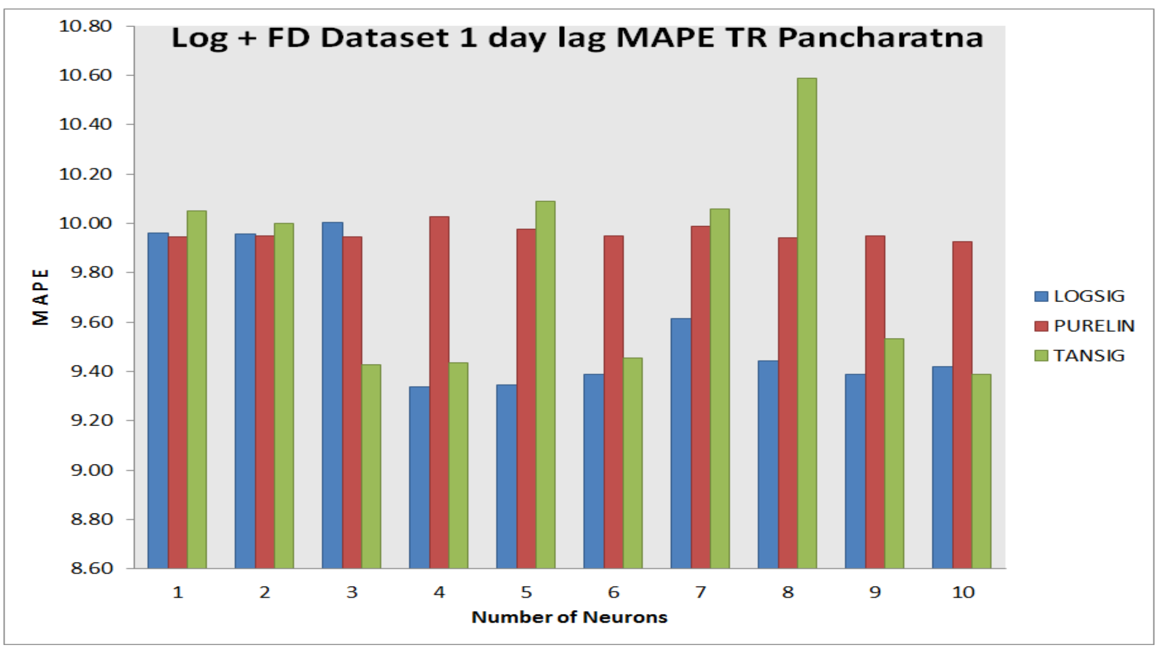


Fig. 4.68 Log + FD Data 1 day lag MAPE TR (Pancharatna)

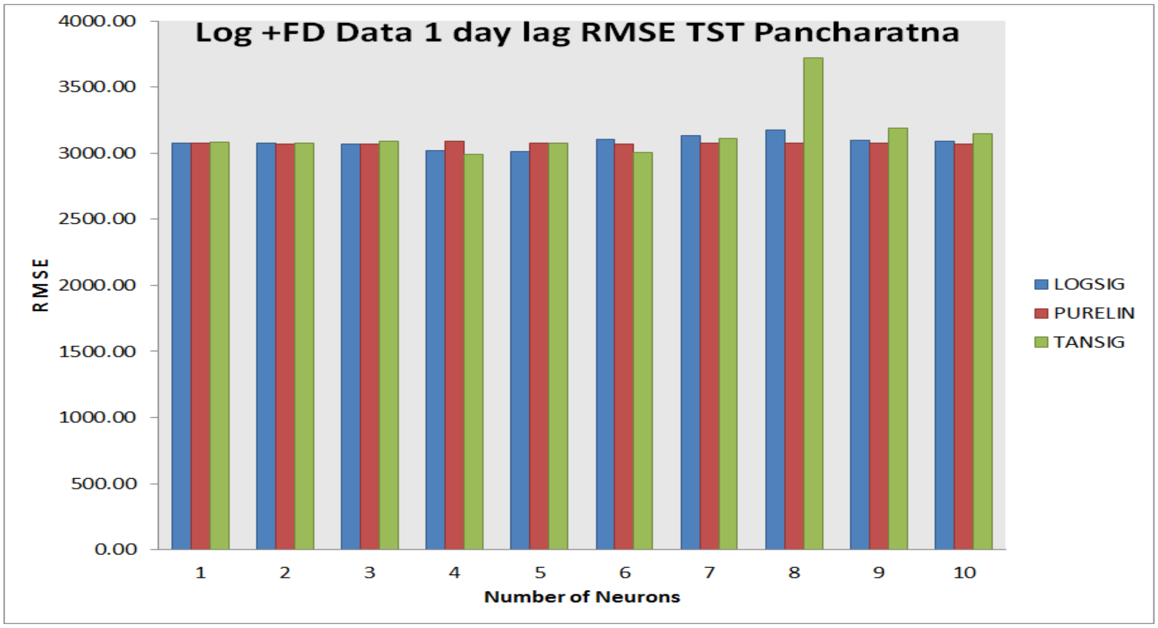


Fig. 4.69 Log + FD Data 1 day lag RMSE TST (Pancharatna)

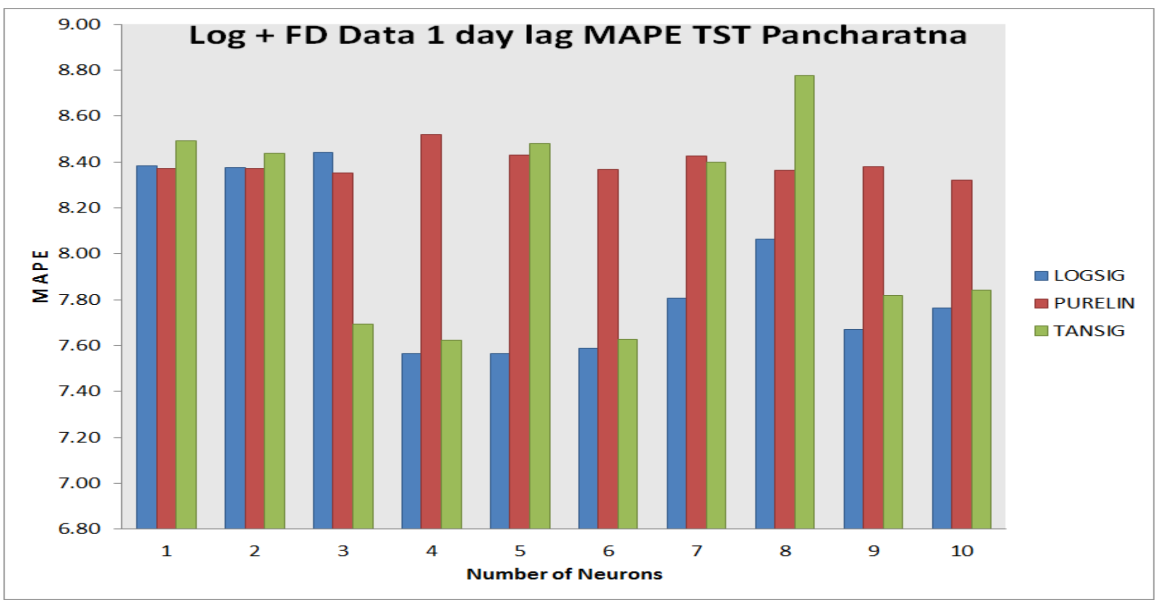


Fig. 4.70 Log + FD Data 1 day lag MAPE TST (Pancharatna)

4.8.8 Log plus First Difference – Two Day Lag

Here the input consists of log + FD data values of two consecutive days and the value for the next day is obtained as the output from the ANN. This value is reconverted to original form and compared with the actual streamflow value to evaluate the error and the required assessmant criteria.

The tables below show the results of these ANN trials and computations.

With two values of inputs in this category the error is slightly reduced. It can be observed again that both LOGSIG and TANSIG perform almost equally well and only marginally than PURELIN type of network. The lowest error reading is highlighted for network of each category.

These results are reproduced as graphics below.

Table 4.54 Log + FD Data 2 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3649.80	9.80	2850.20	8.38
2	3560.40	9.21	2789.75	7.73
3	3579.20	9.22	2812.01	7.73
4	3558.00	9.16	2824.53	7.71
5	3603.60	9.49	2786.66	7.98
6	3460.20	8.72	2929.10	7.40
7	3578.20	9.37	2896.44	8.01
8	3481.80	9.06	2863.94	7.65
9	3588.10	9.08	2816.48	7.51
10	3328.50	8.60	3000.57	7.21

Table 4.55 Log + FD Data 2 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3652.00	9.75	2855.70	8.36
2	3653.20	9.75	2866.50	8.38
3	3654.30	9.73	2860.90	8.34
4	3651.70	9.74	2855.40	8.34
5	3653.60	9.74	2852.00	8.33
6	3652.40	9.69	2857.90	8.27
7	3651.40	9.65	2851.10	8.21
8	3658.50	9.79	2859.90	8.40
9	3649.50	9.80	2861.90	8.44
10	3649.40	9.80	2864.70	8.45

Table 4.56 Log + FD Data 2 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3649.20	9.73	2852.70	8.32
2	3557.60	9.12	2781.80	7.66
3	3576.90	9.24	2818.40	7.74
4	3576.70	9.10	2747.10	7.54
5	3567.70	9.13	2817.20	7.62
6	3371.80	8.72	2525.10	7.01
7	3573.80	9.33	2804.80	7.88
8	3521.60	9.09	2730.50	7.52
9	4257.40	9.90	3507.20	8.58
10	3533.90	9.17	2785.00	7.72

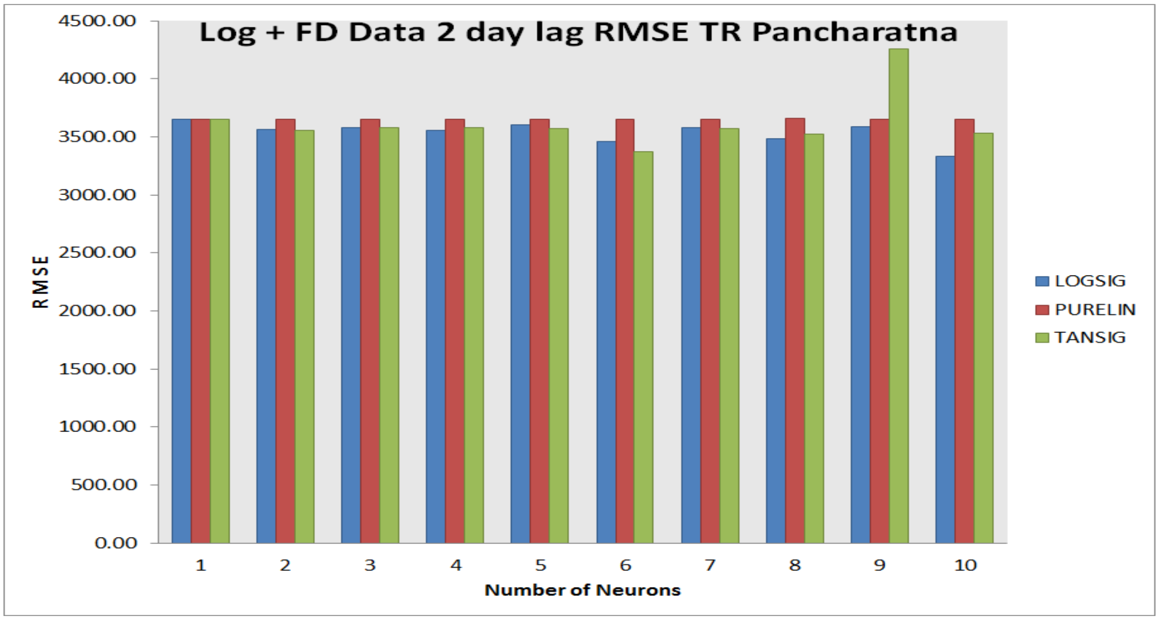


Fig. 4.71 Log + FD Data 2 day lag RMSE TR (Pancharatna)

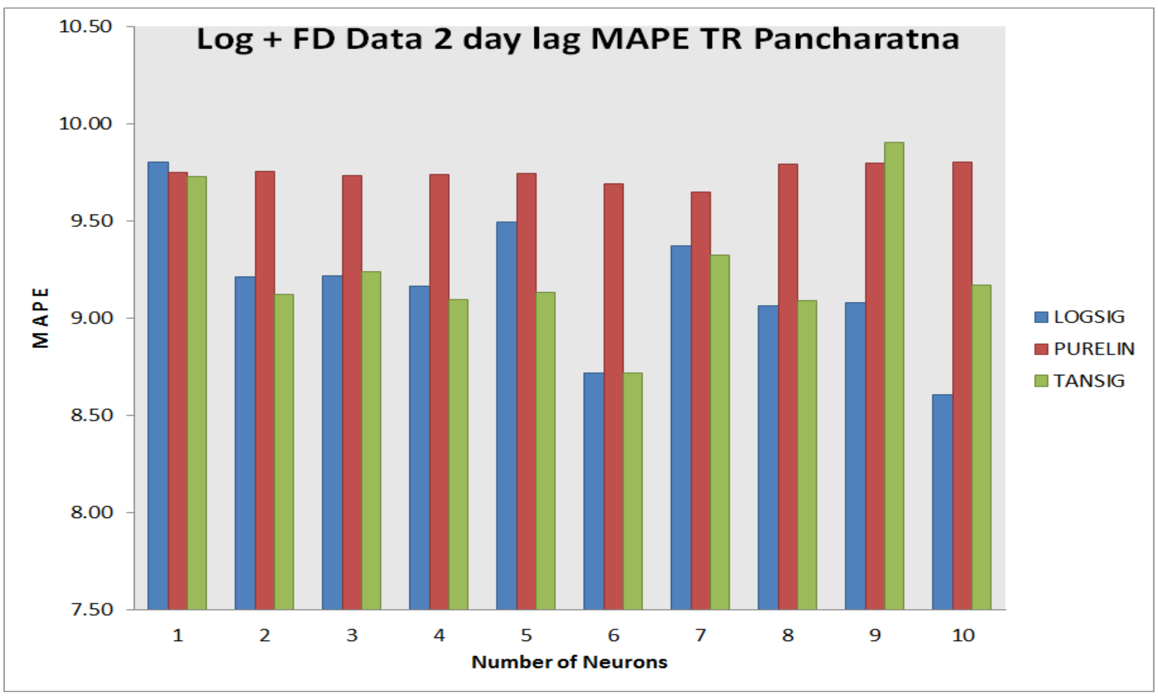


Fig. 4.72 Log + FD Data 2 day lag MAPE TR (Pancharatna)

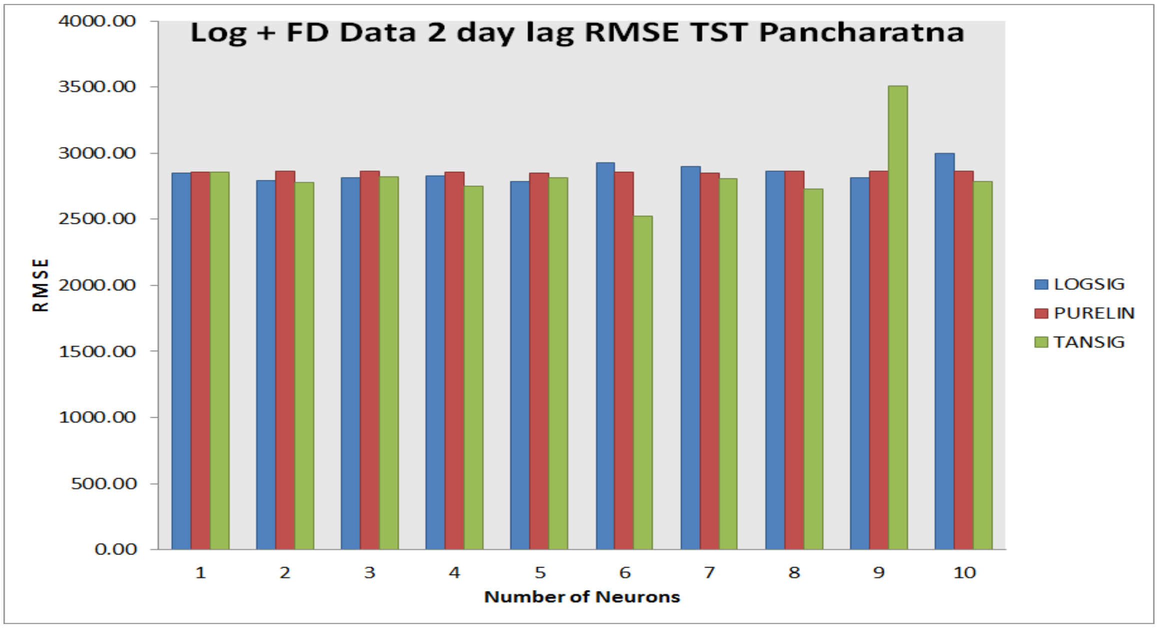


Fig. 4.73 Log + FD Data 2 day lag RMSE TST (Pancharatna)

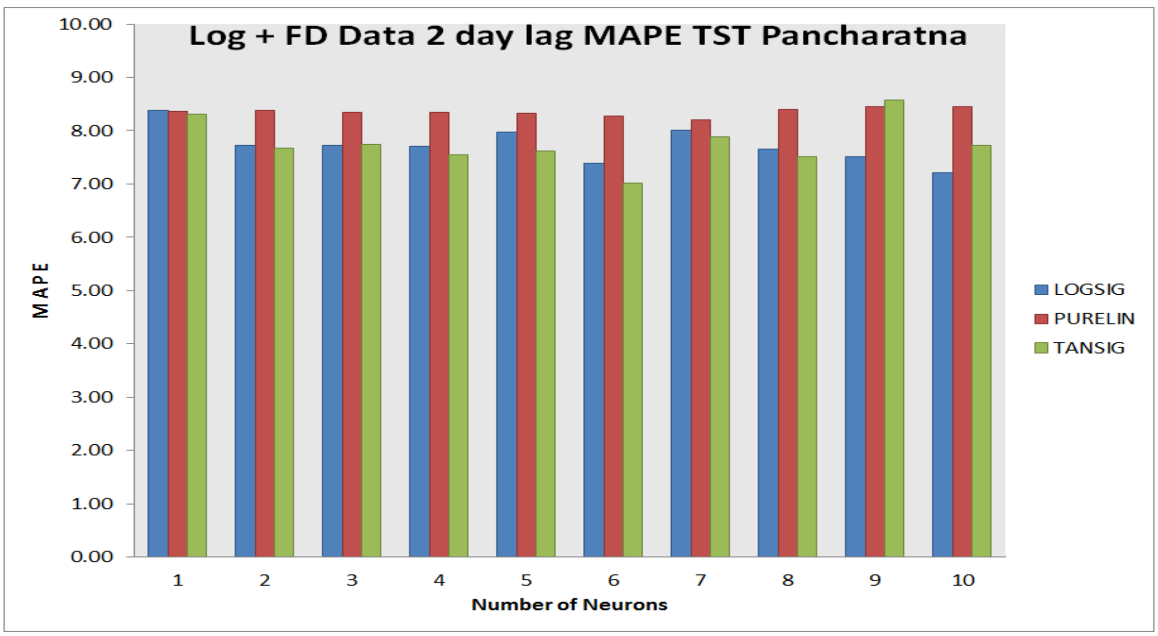


Fig. 4.74 Log + FD Data 2 day lag MAPE TST (Pancharatna)

4.8.9 Log plus First Difference – Three Day Lag

Here the data for three consecutive days from the pre-processed dataset of 'Log plus First Difference' is given to the ANNs as input and the predictions of the next day are obtained as output. The output is post-processed to bring it to original format and then these predictions of streamflow are compared with actual values to assess the evaluation criteria.

The tables below show the results of these computations.

It can be seen that the error is further reduced by three day lag, i.e. providing data of three consecutive days to the networks. Here LOGSIG and TANSIG perform almost equally well with marginal advantage over PURELIN type of network. The low values of each category are highlighted.

The results are also visually represented.

Table 4.57 Log + FD Data 3 day lag – LOGSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3639.10	9.79	2854.89	8.42
2	3560.50	9.29	2741.32	7.78
3	3563.90	9.25	2760.11	7.80
4	3577.70	9.25	2797.15	7.74
5	3748.10	9.19	2811.57	7.78
6	3446.80	8.82	2725.41	7.06
7	3581.10	9.32	2832.62	7.95
8	3292.80	8.60	2630.85	6.83
9	3572.30	9.02	2901.30	8.05
10	3541.90	9.21	2855.96	7.90

Table 4.58 Log + FD Data 3 day lag – PURELIN

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3660.90	9.74	2858.10	8.33
2	3652.30	9.76	2862.20	8.40
3	3652.50	9.76	2863.90	8.39
4	3653.50	9.75	2861.10	8.37
5	3651.00	9.77	2854.30	8.39
6	3652.60	9.69	2851.90	8.26
7	3652.10	9.84	2862.60	8.49
8	3648.60	9.76	2855.80	8.38
9	3650.70	9.79	2858.10	8.42
10	3653.50	9.82	2870.00	8.49

Table 4.59 Log + FD Data 3 day lag – TANSIG

Number Of Neurons	RMSE TR	MAPE TR	RMSE TST	MAPE TST
1	3642.20	9.77	2849.70	8.38
2	3646.10	9.85	2858.20	8.45
3	3551.90	9.21	2782.60	7.79
4	3578.10	9.31	2827.20	7.92
5	3550.00	9.54	2752.70	8.17
6	3556.50	9.17	2749.40	7.73
7	3588.40	9.06	2804.70	7.56
8	3550.70	9.47	3487.30	8.57
9	3395.40	8.56	2710.50	6.85
10	3541.10	9.10	2731.50	7.53

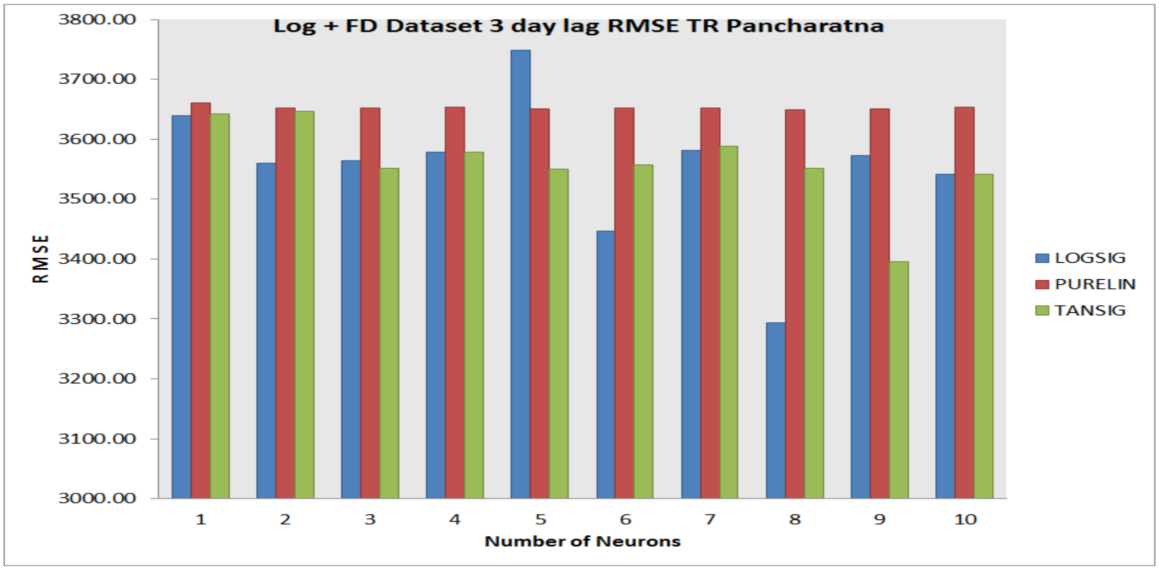


Fig. 4.75 Log + FD Data 3 day lag RMSE TR (Pancharatna)

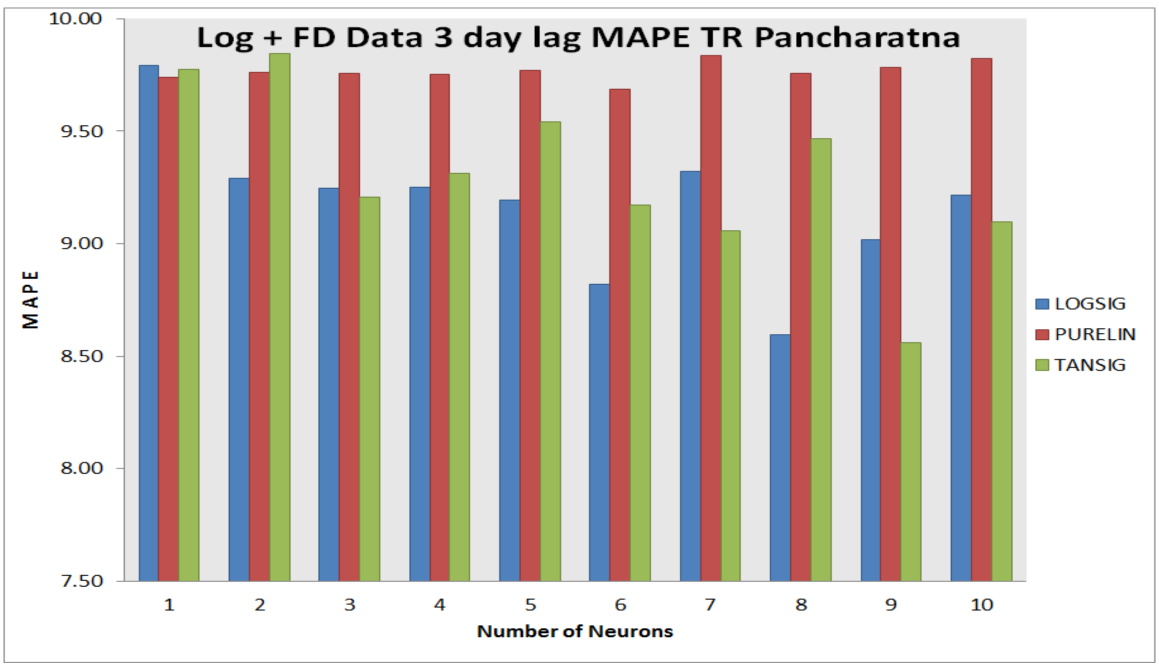


Fig. 4.76 Log + FD Data 3 day lag MAPE TR (Pancharatna)

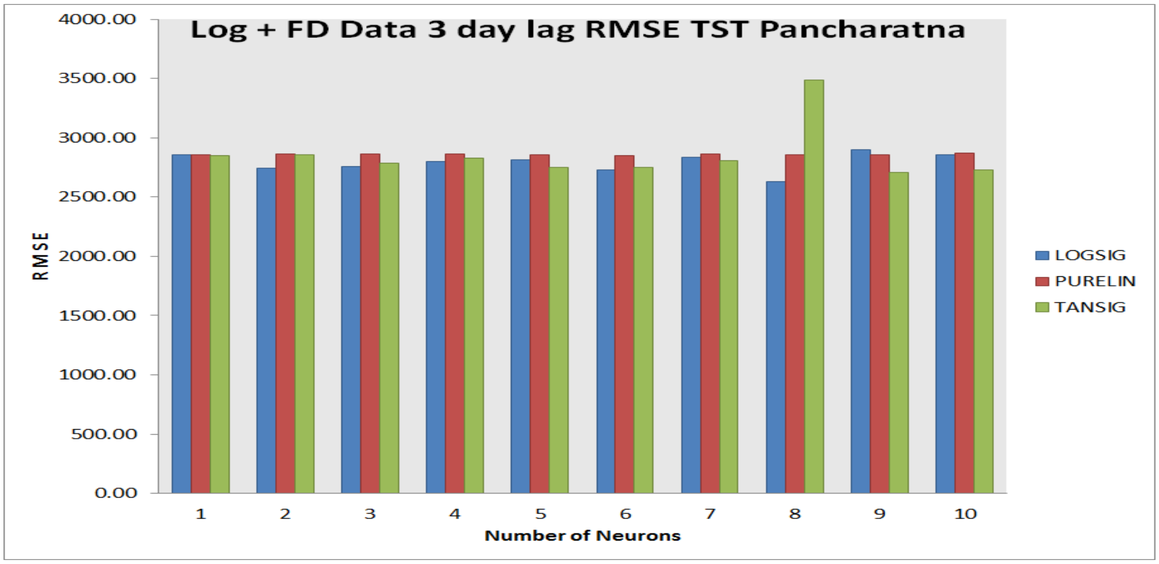


Fig. 4.77 Log + FD Data 3 day lag RMSE TST (Pancharatna)

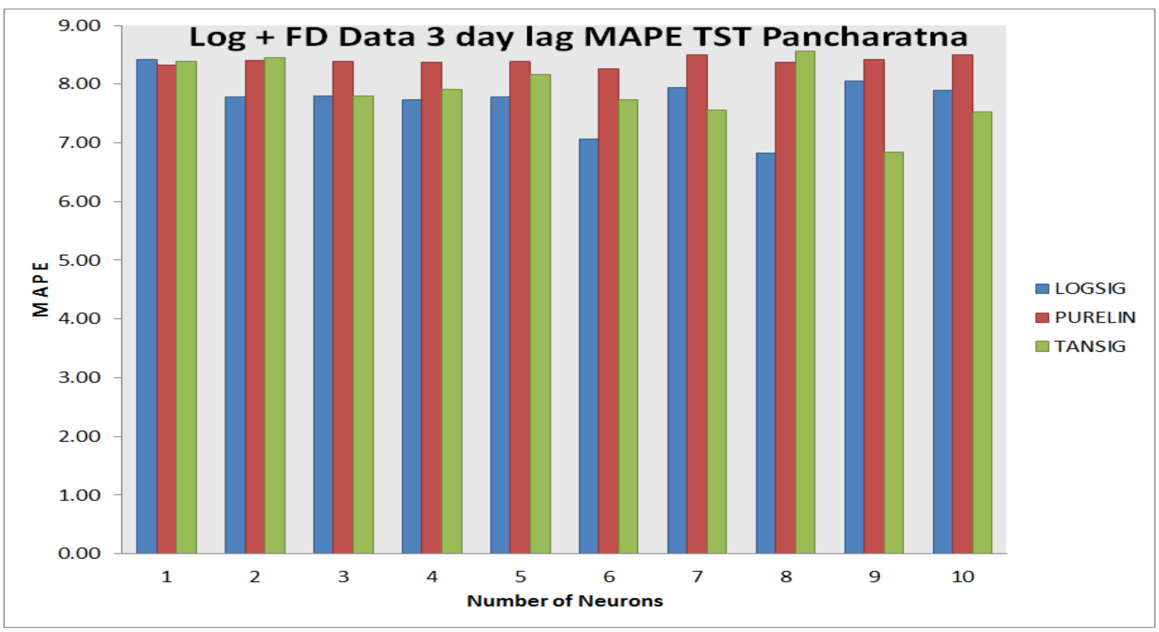


Fig. 4.78 Log + FD Data 3 day lag MAPE TST (Pancharatna)

4.9 Selection of Network – Dataset Combination

Thus in each type of datasets the best performance criteria are gathered together for the testing dataset as indicated in the Table no. 4.79. In Log Transformed Dataset, there are many networks very close to each other and the selection is not based only on lowest value of MAPE in the testing dataset, but its consistency is also taken into account.

Table 4.60 Network Selection Testing Dataset (Pancharatna)

Dataset	No. of Lagged Terms	Best Network Structure	LOGSIG		PURELIN		TANSIG	
			RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)	RMSE (m^3/s)	MAPE (%)
Raw	1	1 – 5 – 1	1480.4	4.38	2845.6	29.7	1503.3	4.48
	2	2 – 7 – 1	1446.8	4.30	2841.0	29.9	22526	84.30
	3	3 – 10 – 1	1310.2	3.9	2831.1	29.6	1365.2	3.80
Log Transformed	1	1 – 3 – 1	1460.1	4.06	1736.3	5.98	1626.1	6.00
	2	2 – 10 – 1	1411.5	3.72	1697.8	5.87	1391.4	3.52
	3	3 – 5 – 1	1337.9	3.46	1673.2	5.88	1397.9	3.96
Log + First Difference	1	1 – 5 – 1	3010.5	7.56	3079.2	8.43	3073.7	8.48
	2	2 – 6 – 1	2929.1	7.40	2857.9	8.27	2525.1	7.01
	3	3 – 8 – 1	2630.8	6.83	2855.8	8.38	3487.3	8.57

Similarly, for training and validation dataset, the results from the different networks are collected together in Table no. 4.80.

Table 4.61 Network Selection Training Dataset (Pancharatna)

Data	No. Of Inputs	Best Network	LOGSIG		PURELIN		TANSIG	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Raw	1	1-5-1	1826.00	5.02	3179.80	34.20	1813.80	5.15
	2	2-7-1	1834.80	5.20	3178.50	34.40	22610.00	81.30
	3	3-10-1	1701.70	4.80	3178.10	34.00	1683.70	4.70
Log	1	1-3-1	1842.80	4.73	2274.70	6.74	1988.10	6.70
	2	2-10-1	1745.70	4.49	2248.30	6.62	1732.20	4.36
	3	3-5-1	1717.80	4.30	2211.00	6.61	1824.80	4.55
Log + FD	1	1-5-1	3739.90	9.35	3827.40	9.98	3814.30	10.09
	2	2-6-1	3460.20	8.72	3652.40	9.69	3371.80	8.72
	3	3-8-1	3292.80	8.60	3648.60	9.76	3550.70	9.47

Thus through trial and error procedure by analyzing 540 trials, 270 for training and validation datasets and 270 for testing datasets, the **Log Transformed Dataset** and the **LOGSIG Network architecture with 3 inputs and 5 neurons in the hidden layer** is found to be working in a very stable way than the Raw Dataset as well as the Log Transformed plus First Difference Dataset for the data at Pancharatna gauging station. The PURELIN architecture is found to give highly non-uniform results and shows high values of errors.

The final choice of network and data type is highlighted in both the training and testing datasets.Thus the final choice is :

Dataset : Log Transformed Dataset with Three Days Lag

Network Architecture : LOGSIG

Network Structure : 3 – 5 – 1

4.10 Testing Consistency of Selected Network

The consistency and robustness of the selected network is tested in three ways here.

1. Performance for high values
2. Performance for low values
3. Performance by interchanging Training and Testing Datasets.

4.10.1 High Values and Low Values

Since the values of streamflow vary from 1723 m^3/s (minimum) to 76236 m^3/s (maximum), the average being 17904 m^3/s, and noting that the values above 45000 m^3/s occur rarely, fixing >30000 m^3/s as the limit for high values, the filter is applied to all values together for the selected network and the RMSE and MAPE are computed.

Fixing the limit for low values as < 5000 m^3/s, the filter is applied to all values predicted by the selected network and the RMSE and MAPE are computed. Following table shows the results.

Table 4.62 Consistency Test High and Low values (Pancharatna)

HIGH VALUES (>35000)		LOW VALUES (<5000)	
R M S E	M A P E	R M S E	M A P E
3518.14	5.10	230.73	2.83

4.10.2 Swapping The Training and Testing Datasets

Here the beginning 1/3rd data points, i.e. 1 to 2434 are taken as the testing dataset and end 2/3rd datapoints i.e. 2435 to 7302 are taken as the training dataset. All the datasets thus created are transformed to Logarithm of the value to the base 10. A new network of LOGSIG type with 6 neurons is created and trained with the training input and target. Then the validation is done with the training dataset without providing the target and the values predicted by the trained ANN are gathered as output for the validation. Testing dataset, also Log-Transformed is fed as input and results are obtained. The results of both the datasets are then compared with the actual values and RMSE and MAPE are computed. The statistical characteristics of the swapped data set and the performance is shown in the next two tables.

Table 4.63 Statistical Characteristics of Swapped datasets

Streamflow Value Q m^3/s	Training Dataset	Testing Dataset	All Dataset
Minimum	1723	2086	1723
Maximum	76236	72914	76236
Average	16270	16984	16503

Table 4.64 Performance of Swapped Datasets

Condition	RMSE TR	MAPE TR	RMSE TST	MAPE TST
After Swapping	1419.25	3.58	2005.90	5.15
Before Swapping	1717.80	4.30	1337.90	3.46

Here we see promising agreement between the results even when the datasets are interchanged validating the idea that if sufficiently large data with appropriate pre-processing technique is presented to a suitable ANN, that ANN can discern the datapattern and with the datapattern as only reference, can predict or forecast the future data with high accuracy, which may be difficult to achieve with the statistical models.

4.10.3 Effect of number of Neurons

As the number of neurons in the hidden layer is increased, the accuracy increases upto a certain stage. After this the increase in the number of neurons affects the accuracy only marginally and most of the times, decreases it. Thus there seems to be an optimum for each network type and datatype combination. In case of log transformed dataset, after increasing the number of neurons to say 4 or 5, the values are so close to each other that a plateau seems to have reached for the reduction of error.

4.10.4 Effect of Number of Inputs or Lagged Terms

As the number of inputs is increased from one to three, almost invariably it is seen that the accuracy of prediction goes on increasing for all data types – network type combimations.

The raw dataset seems to give incosistent results. In some cases of raw data combined with PURELIN activating functions, one input seems to give less error than two or three inputs. This is true for only the above mentioned combination for two cases and it can be treated as an exception.

It can be stated that 3 input networks indeed give better results. As a trial, four and five input networks were tried before deciding the plan of work, as stated earlier, but these do not seem to improve the results any further. In some trials, even the accuracy diminished drastically showing a possibility of overtraining tendency due to data overload.

4.10.5 Plotting the Predicted Values – Comparison with Actual Values

The following plot shows the entire data as forecast with the selected network dataset combination in comparison with the actual data recorded at the guaging station Pancharatna. For better resolution, the plot is divided in four parts.

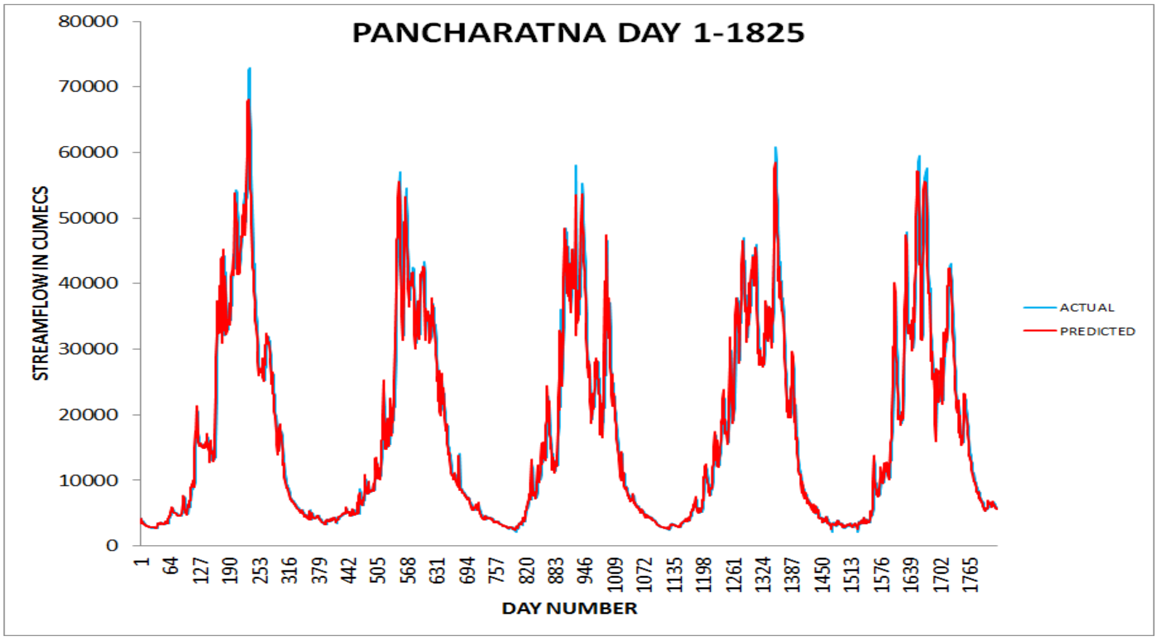


Fig. 4.79 Pancharatna Comparison of predicted and actual values Day 1-1825

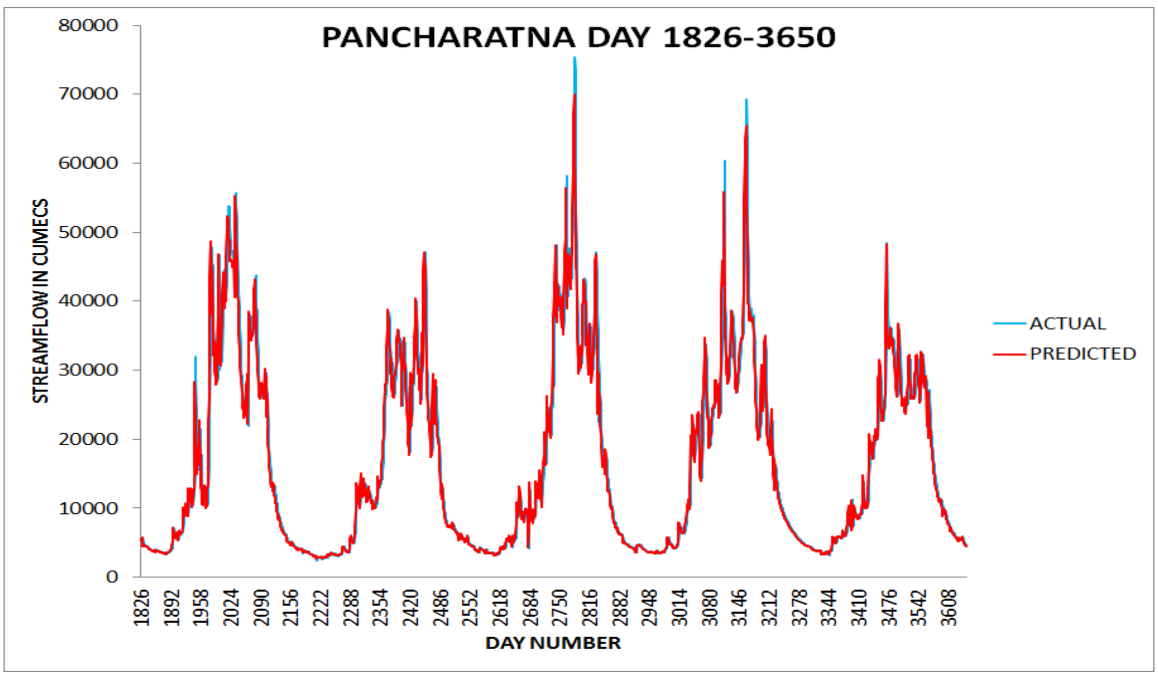


Fig. 4.80 Pancharatna Comparison of predicted and actual values Day 1826-3650

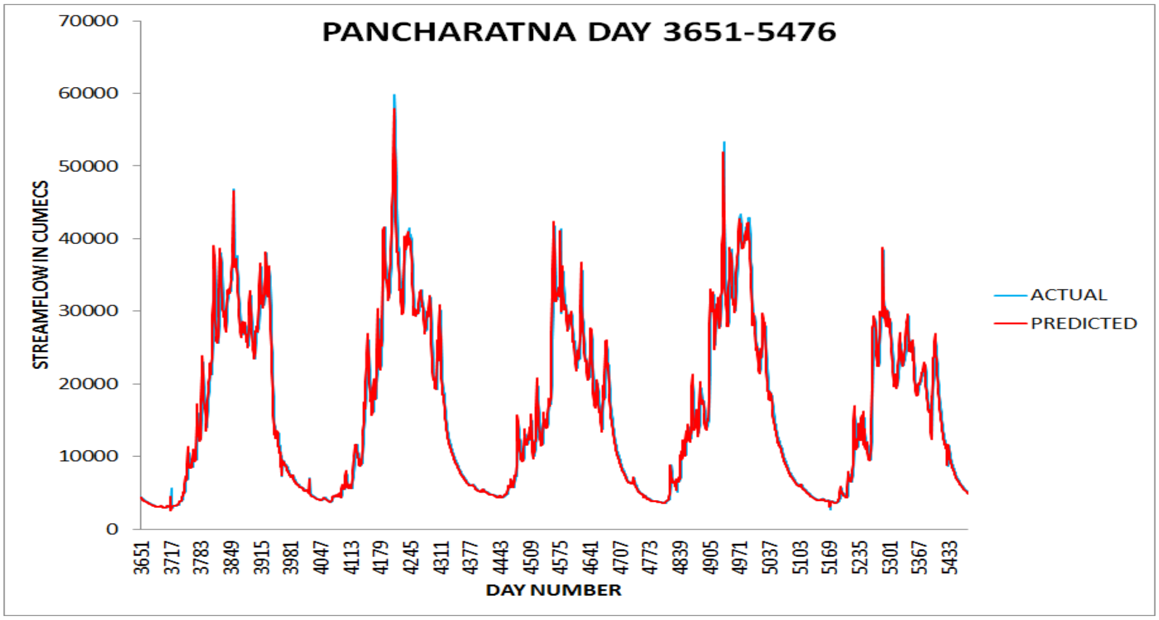


Fig. 4.81 Pancharatna Comparison of predicted and actual values Day 3651-5476

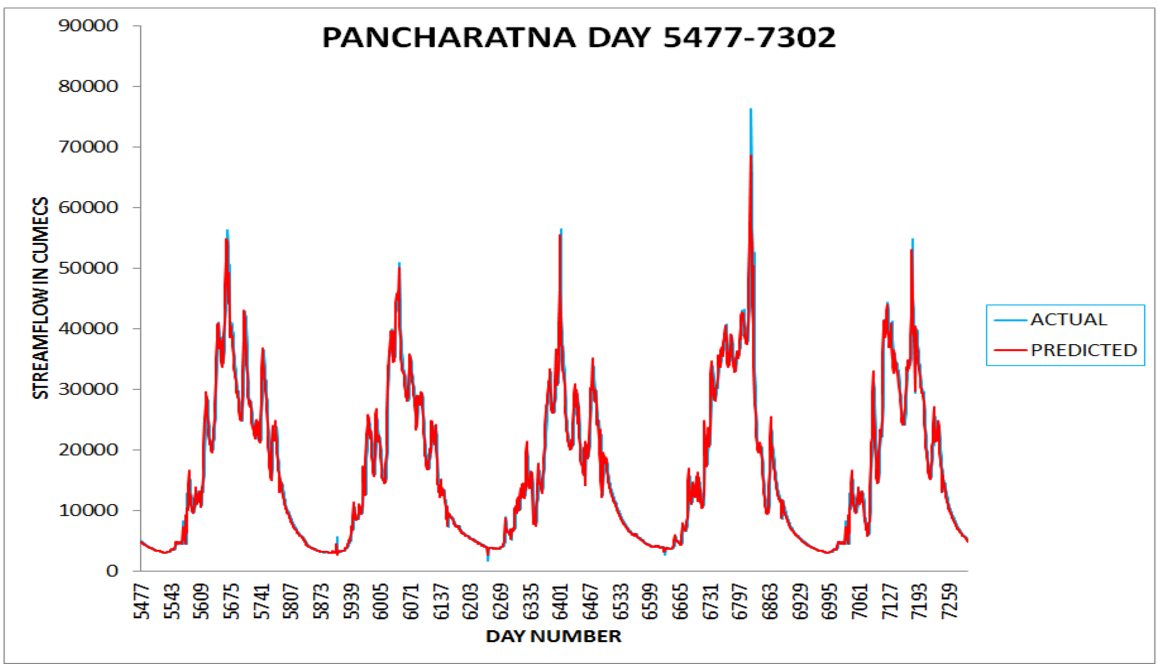


Fig. 4.82 Pancharatna Comparison of predicted and actual values Day 5477-7302

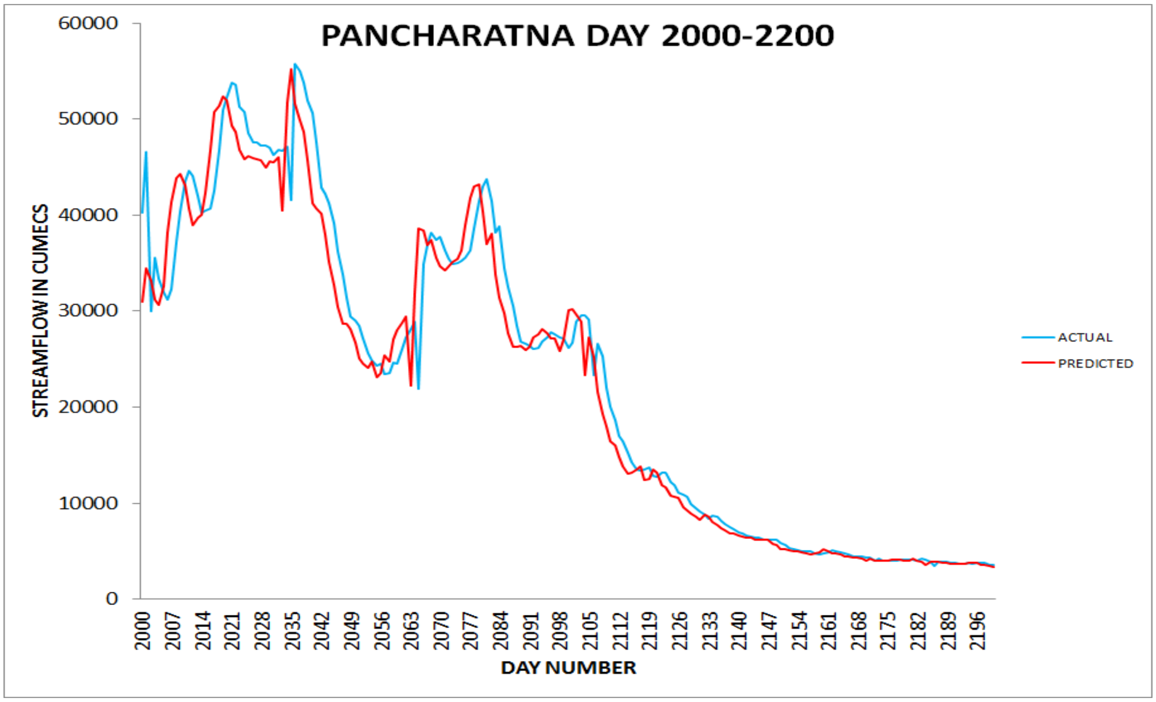


Fig. 4.83 Pancharatna Day 2000-2200 Enlarged view

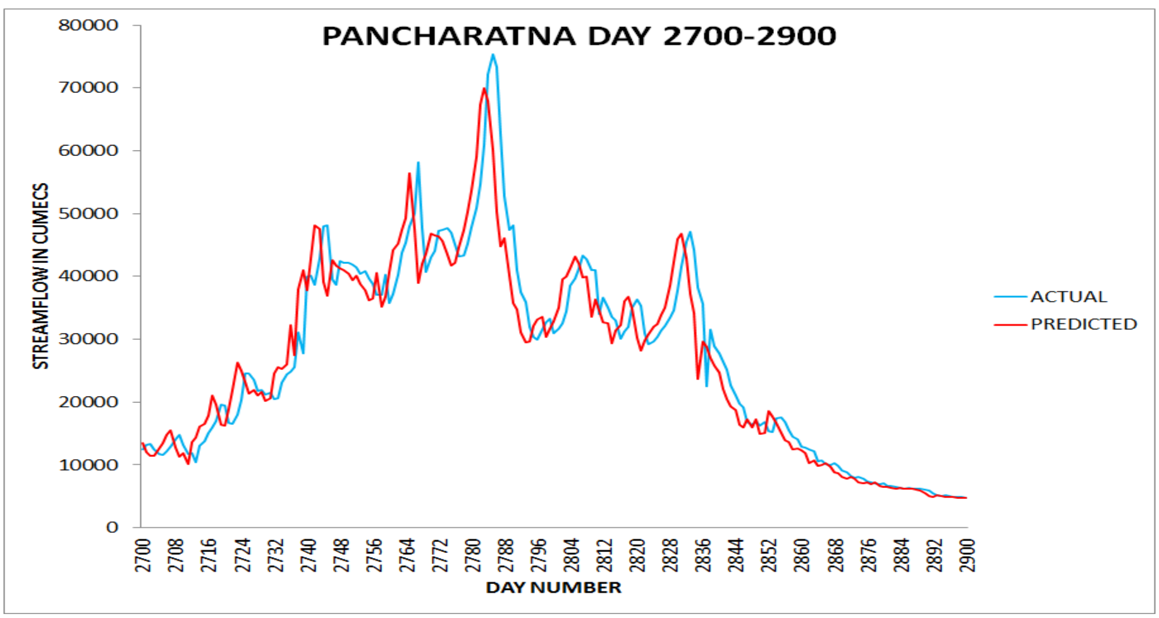


Fig. 4.84 Pancharatna Day 2700-2900 Enlarged view

Close agreement between the predicted and actual values is observed. Due to large data volume the discrepancies are not clearly seen but the part where the discrepancy is apparent is plotted with high resolution in the individually enlarged representive plots.

4.10.6 Comparison of Results for Pandu and Pancharatna

With all the trials conducted for various dataset-network combinations, almost identical combinations are identified for both the stations. Since the river is the same and the two stations are approximately 150 km away i.e. not too far from each other; this results fortifies the validity of the selection.

The very low values of MAPE and RMSE are most promising for using this technique in actual practice for predicting and preparing a streamflow forecast in advance especially just before the usual flooding season or when the monsoon starts to set in.

Following table gives the comparison of results at the two stations.

Table 4.65 Comparison of results of Pandu and Pancharatna

Station	Data Type	Best Network Structure	Activation Function	Testing Dataset		Training Dataset	
				RMSE m^3/s	MAPE %	RMSE m^3/s	MAPE %
Pandu	Log Data	3-6-1	LOGSIG	907.95	2.12	1002.80	2.67
Pancharatna	Log Data	3-5-1	LOGSIG	1337.9	3.46	1717.8	4.30

The higher value of error seen at pancharatna can be attributed to some of the following factors:

1. Skewness coefficient at Pancharatna is more.

2. The data range is wider at Pancharatna.
3. Between Pandu and Pancharatna more than 100 tributaries join the main stem of Brahmaputra River and 20 of theses are big streams themselves.
4. Channel at Pandu is constrained with about 2.5 km width whereas the channel at Pancharatna is 5 to 7 km wide.
5. The planform at Pandu is constant but at Pancharatna island formation and braiding occurs during dry season which may give rise to discrepancy of data.

4.11 Conclusion

It is seen that right combination of Data Preprocessing technique, number of neurons in the hidden layer, number of inputs and the activation function can model streamflow time series effectively with reasonable accuracy with only streamflow data as the variable.

CHAPTER – 5

SUMMARY AND CONCLUSION

The present study explores the potential and suitability of ANN methods by using data preprocessing techniques in time series river flow forecasting at two gauging stations of Brahmaputra River in Assam part of India. This chapter summarizes the initiatives that have been taken in order to achieve the objectives of the study. Finally, some suggestion for future work has also been included at the end of the chapter.

Designing an ANN model for time series forecasting involves a large set of parameters especially for network training and topology. Due to their flexibility ANN lacks of systematic procedure for model building. Therefore obtaining a reliable neural network model involves selecting a large number of parameters experimentally through trial and error. Feedforward ANN training is usually not very stable since the training process may depend on the choice of a random start. Training is also computationally expensive in terms of training time used to determine the appropriate network structure. The degree of success, therefore, may fluctuate from one training pass through another.

Hence, an empirical study has been done to search for optimal network architecture, activation function and data preprocessing technique for the time series river flow forecasting. We present different preprocessing techniques for removing nonstationary and evaluated their properties by producing one step ahead forecasts. We also examined three types of ANN models using Logsig, Tansig and Purelin activation functions. Also, the number of hidden nodes are varied from 1 to 10 for various input combinations to obtain the optimized output.

5.1 Summary of work

The current study, the following tasks has been carried out at different stages-

The daily river flow time series data for two stations such as Pandu(u/s) and Pancharatna (d/s) have been collected for a period of 20 years (1980-1999). Here, First 67% (2/3) of observed data are used for model calibration and remaining 33% (1/3) data are used for validation. A huge number of ANN models are generated on the basis of combining number of inputs, number of outputs, number of hidden neurons.

The models are evaluated with validation set through two forecasting accuracy measures: Root Mean Squared Error (RMSE) and (MAPE) Mean absolute Percentage Error. These are used to evaluate the forecasting performance accuracy of developed models. Following tasks have been carried out in the study:

- Three datasets are derived from three data preprocessing techniques.
- Various data matrices are generated for three sets of data on the basis of lagged terms such as one day lag , two day lag and three day lag input flow data that would act as input and the one step ahead forecast that would act as the network output.
- Each sample datasets is subdivided into two sets: ‘training and validation set’ and ‘test set’ in order to obtain a network which is capable of generalizing and performing well with new cases.
- Total 270 (3 input scenarios due to three lagged datax3different data preprocessing sets x 3 different activation functions x 10 hidden neurons such as 1 to 10 variation of nodes) ANN models were developed involving MLP network through the BP learning algorithm for every station .

  a] Keeping learning rate and momentum coefficient constant in all the training such as 0.5 and 0.5 respectively.

b] The numbers of iteration are kept fixed at 40000

- Evaluation and selection of ANN models-

  a] Based on testing performance, optimal ANN model which presents the best forecast result is selected from each datasets.

  b] The forecast results produced by optimal ANN model from each datasets are compared in order to select the appropriate data preprocessing technique.

  c] Dataset which produces the best forecast result is selected as the proper data preprocessing technique for time series river flow forecasting.

- Experimental results are analyzed and discussed.

5.2 Contribution

Following are the contributions from this study:

- Data series that has been preprocessed with log transformation is able to accelerate the convergence rate and produce better forecast result.
- Sigmoidal activation function such as LOGSIG model generates better learning and forecast capability when compared to TANSIG and PURELIN model in most of the experiments.
- The proposed LOGSIG model is able to improve the convergence problem with small number of hidden nodes.
- Network with three day lagged inputs generates better forecasting performance.

5.3 Conclusions

Following are the conclusions drawn from this study:

- Employing an appropriate data preprocessing technique is highly beneficial to simplify the ANN training and improve the forecasting quality in this study.
- For the prediction of streamflow at Pandu station, the ANN with LOGSIG activating function, Log Transformed dataset, 6 neurons in the hidden layer is found most efficient, whereas at Pancharatna, similar network with 5 neurons in the hidden layer is found most efficient.
- Log Transformation as a preprocessing technique for a time series of large variance and non-stationary nature is a very effective tool to improve the prediction by Artificial Neural Networks.
- For a non- linear dataset the activation function PURELIN does not perform well whereas LOGSIG is found to be most suitable for non- linear datasets.
- The number of neurons in the hidden layer of the ANN affect the performance. The accuracy increases due to increasing the number upto a certain extent only after which a plateau is reached and the performance remains unaffected or may decrease in some cases if the number of neurons is further increased.
- More number of inputs improves the performance of the ANNs upto a certain extent after which the increase in inputs may not have any effect on the performance.
- Incorporating the first difference of successive terms in the input for a time series analysis does not improve the accuracy of ANNs and in fact may decrease it.
- Further study will be required for computational time requirement, flexibility, limited data, limited input variables and simplicity to user for concrete conclusion.

5.4 Limitations

- Here, the selective data preprocessing techniques such as logarithmic transform provides better results than another Log+First difference data set. Other data preprocessing techniques like wavelet transformation, seasonal difference, and logarithmic return could not be tried due to time constraints.

- The model has been tested for one step lead-time only which is sufficient for short term management. For long-term planning, multistep lead-time forecasting is required.
- Other factors which are not included but supposed to be influential such as rainfall, seepage, infiltration, evapotranspiration, temperature, catchment characteristics, geomorphologic properties in the network inputs could explain the poor relationship between the persistence and future river flow.

5.5 Scope for future work

The following are some suggestions that could lead to the improvement of the results that have been obtained and some possible points that could lead to future research.

- Applying different sample size of training set, validation set and testing set may be tried. Appropriate data sampling may help in order to avoid overfitting problem that occurs in this study.
- In order to improve the forecast performance of ANN model, more trial and error experiments need to be done on other network parameters that are not studied in this research for example varying the momentum term and the learning rate to accelerate the BP learning process.
- Further, to optimize the internal parameters of the networks and to enhance the forecasting accuracy in real field situation of developing countries, integration of other techniques like Fuzzy Logic, Genetic programming may be employed.

REFERENCES

ASCE Task Committee, (2000a). "Artificial neural networks in hydrology-I: Preliminary concepts." *J. Hydrologic Engineering, ASCE,* 5(2), 115–123.

ASCE Task Committee, (2000b). "Artificial neural networks in hydrology-II: Hydrologic applications." *J. Hydrologic Engineering, ASCE* 5(2), 124–137.

BaneerjeePallavi., Prasad R. K.and Singh V. P. (2009) "Forecasting of groundwater level in hard rock region using artificial neural network," *journal of Environ Geol,* 58, 1239-1246.

Besaw,L.E., Donna M. Rizzo, Paul R. Bierman, William R. Hackett,(2010), "Advances in ungauged stream flow prediction using artificial neural networks", Journal of Hydrology 386 (2010) 27–37.

Bierkens, M. F. P., (1998), 'Modeling water table fluctuations by means of a stochastic differential equation', *Water Resources Research* 34, 2485–2499.

Cannas.B, Fanni.A, See.L.,Sias.G., (2006),"Data preprocessing for river flow forecasting using neural networks: wavelet transforms and data partitioning". Physics and Chemistry of the Earth 31 (18), 1164–1171.

Chen L., and Chen C., Pan. Y. (2010).Groundwater Level Prediction Using SOM-RBFN Multisite Model. *J. of Hydrologic Engineering ASCE,* 624-631.

Christian w. dawson and Robert wilby, (1998), "An artificial neural network approach to rainfall runoff modeling"Hydrological SciencesJournaldes Sciences Hydrologiques, 43(1) February.

Cigizoğlu, H. K. (2005). Application of generalized regression neural networks to intermittent flow forecasting and estimation.*J. Hydrol. Engng ASCE* 10(4), 336–341.

Coulibalya,P and Connely K. Baldwin, (2005), "Nonstationary hydrological time series forecasting using nonlinear dynamic methods", Journal of Hydrology 307 (2005) 164–174.

Dawson,C.W and Wilby,R.L.(2001).Hydrological modeling using Artificial Neural Networks.Progress in physical geography,25(1),pp.80-108

Deka P.C. and Prahlada R (.2012). "Discrete Wavelet Neural Network approach in significant wave height forecasting for multistep lead time" Ocean Engineering ,April'2012, Vol-43pp..32-42.

Deka,P. And Chandramoulli,V.(2005). "Fuzzy neural network modelling for hydrologic flow routing."ASCEJ.of Hydrologic Engineering,2005,July/August,vol.10(4),pp.302-314

Granger,C.W.J(1994).Forecasting in economics,time series prediction-forecasting the future and understanding the past,N.A.-Gershenfeld and A.S.Weigend(eds.)reading M.A.,Addison-Wesley,pp.529-538.

H. KeremCigizoglu and Ozgur Kisi, (2005),"Flow prediction by three back propagation techniques using k-fold partitioning of neural network training data", Nordic Hydrology Vol 36 No 1 pp 49–64 q IWA Publishing.

Hamid, S. A., and Zahid, I. (2004)."Using neural network for forecasting volatility of S&P 500 index futures prices." J. Bus. Res., 57, 1116–1125.

Haykin S (1999). Neural networks: A Comprehensive Foundation Second edition, Prentice-HallEnglewood Cliffs, NJ.

K.K.Phoon, M.N.Islam, C.Y.Liaw and S.Y.Liong,(2002) "Practical Inverse Approach for Forecasting Nonlinear Hydrological Time Series",Journal of Hydrologic Engineering, Vol. 7, No.2, March 1, ©ASCE, ISSN 1084-0699/2002/2-116–128.

Kajitani,Y.,Hipel,K.W.,Mceod,A.I.(2005), "Forecasting nonlinear time series with feedforward neural networks;a case study of Canadian Lynx data".J.of Forecasting,24,105-117.

Keskin,M.E and DilekTaylan, (2009),"Artificial Models for Inter-basin Flow Prediction in Southern Turkey" Vol. 14, No. 7, July 1, 2009. ©ASCE, ISSN 1084-0699/2009/7-752–758.

Kişi, O. (2006b) "Generalized regression neural networks for evapotranspiration modeling," Journal Hydrol. Sci. 51(6), 1092-1104.

Kong,J.H.L. and Martin,G.P.M.D.,(1995) "A backpropagation neural network for sales forecasting.IEEE,2121-2124.

Kumar,A.M andJain,A.(2007)"Hybrid neural network models for hydrologic time series forecasting", Elseivier, Applied Soft Computing 7, 585-592.

Karunanithi, N., Grenney, W. J., Whitley, D. &Bovee, K. (1994). " Neural networks for river flow prediction". *J. Computing in Civil Engg.***,** ASCE, 8(2), 201-220.

Lam,M.,(2004) "Neural network techniques for fincial performance prediction:integrating fundamental and technical analysis".Decision Support Systems,37,567-581.

Legates, D. R.; McCabe, Jr., (1999).Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. Water Resour. Res., 35 (1), 233-241

Legates, D. R.; McCabe, Jr., (1999).Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. Water Resour. Res., 35 (1), 233-241

Lin, G. F., and Chen, L. H. (2005). “Time series forecasting by combining the radial basis function network and the self-organizing map.” Hydrolog.Process., 19(10), 1925–1937.

Lopes,M.L.M,Minussi,C.R. and Lotufo,A.D.P. “A fast electric load forecasting using neural networks.”In:proc.43rd IEEE Midwest Symp. On Circuits and Systems,Learning MI.8-11,August,2000,IEEE,1-4

Ma Lishan., Yuan Dekui., Tao Jianhua., Yang Guoli and Sun Yong (2009). “Prediction of groundwater level based on DE-BP neural network,” journal of Environmental Technology and Engineering, 2 (1), 10-15.

Maier H. R. and Dandy G. C., (2000). “Neural Networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications, Environmental Modeling and Software, Elsevier, 15, 101-124.

Maier, H. R. and Dandy, G. C. (2000).‘‘Neural networks for the prediction and forecasting of water resources variables:A review of modelling issues and applications”.EnvironmentalModelling& Software,15,101-124.

Maier, H. R. and Dandy, G. C., (1997), ‘Determining inputs for neural network models of multivariate time series’, Microcomputers in Civil Engineering 12, 353–368.

Mehmet C. Demirel, AnabelaVenancio ,ErcanKahya, (2009), “Flow forecast by SWAT model and ANN in Pracana basin, Portugal”, Advances in Engineering Software 40, 467–473.

Moradkhani, H., Hsu, K., Gupta, H., Sorooshian, S., (2004). “Improved streamflow forecasting using self-organizing radial basis function artificial neural networks”. Journal of Hydrology 295, 246–262.

Nag, A. K., and Mitra, A. (2002). “Forecasting the daily foreign exchange rates using genetically optimized neural networks.” J. Forecast., 21, 501–511.

Nayak, P. C., SatyajiRao Y. R., and Sudheer K. P. (2006). “Groundwater Level Forecasting in a shallow Aquifer Using Artificial Neural Network.” J. Water Resource Management 20, 77–90.

Nelson, M.Hill, T, Remus, T., O’Connor, M (1999). “Time series forecasting usingneuralnetworks:Shouldthedatabe deseasonalisedfirst?”J.Of Forecasting,18;359-367.

Nelson, M.Hill, T, Remus, T., O’Connor, M (1999). “Time series forecasting usingneuralnetworks: Shouldthedatabe deseasonalisedfirst?”J.Of Forecasting,18;359-367.

neural transfer functions”.Neural computing surveys 2,163-212.

Nguyen,H.H. and Chan C.W.(2004). “A comparison of data preprocessing strategies for neural network modelling of oil production prediction.”In:proceedings of the third IEEE International conference and cognitive informatics(ICCI’04).IEEE Computer Science.

Nourani, V., Alami, T Mohammad, Aminfar, Mohammad.H (2009).“A combined neural-wavelet model for prediction of Ligvanchai watershed precipitation.” Elsevier, Engineering Applications of Artificial Intelligence, 22, 466–472.

Özgür Kisi, (2007), “Development of Streamflow-Suspended Sediment Rating Curve Using a Range Dependent Neural Network”, International Journal of Science & Technology Volume 2, No 1, 49-61, 2007.

Panda Dileep K, Mishra A Jena S. K., James B.K. and Kumar A (2007) “The influence of drought and anthropogenic effects on groundwater level in Orissa, India,” journal of Hydrology, 343, 140-153.

Plummer,E.A(2000). “Time series forecasting with feedforward neural networks:Guidelines and limitations”.MasterThesis,University of Wyoming.

Powell, M. J. D. (1987). “Radial basis functions for multivariable interpolation: A review.” Algorithms for approximation, J. C. Mason and M. G. Cox, eds., Clarendon, Oxford, U.K., 143–167.

Sajikumar, N., and Thandaveswara, B.S., (1999). “A non-linear rainfall–runoff model using an artificial neural network”. J. Hydrol. 216, 32–55.

Scanlon BR, Healy RW, Cook PG (2002) Choosing appropriate technique for quantifying groundwater recharge. Journal of Hydrology Vol 10, pp 18-39.

Sethiet, A.Kumar S.P. Sharma and H.C. Varma (2010). “Prediction of water table depth in hard rock basin by using artificial neural network”, International journal of water resources and environmental engineering Vol.2 (4), p 95-102, June 2010.

Sharma, J.N.(2005) “Fluvial process and morphology of the Brahmaputra river in Assam,India”.*Geomorphology*,70,226-256.

Shirmohammadi, A., I. Chaubey, R. D. Harmel, D. D. Bosch, R. Munoz-Carpena, C. Dharmarsi, A. Sexton, M. Arabi, M. L. Wolfe, J. Frankenberger, C. Graff, and T. M. Sohrabi. (2006). Uncertainty in TMDL Models. Trans. ASABE, 494: 1033-1049.

Singh R. M. and B. Datta (2007). “Artificial neural network modeling for identification of unknown pollution sources in groundwater with partially missing concentration observation data.” J. Water Resource Management, (21), 557–572.

Sreenivasulu,D. and Deka,Paresh Chandra.(2011) “A comparative study on RBF and NARX based methods for forecasting of groundwater level”.Int. J. Earth science and Engg.,vol.04(4),August,743-756.

Sudheer, K. P., Gosain, A. K., and Ramasastri, K. S., (2002), “A data-driven algorithm for constructing artificial neural network rainfall-runoff models,” Hydrological Processes 16, 1325–1330.

SurinderDeswal, and Mahesh Pal, (2008), “Artificial Neural Network based Modeling of Evaporation Losses in Reservoirs”, World Academy of Science, Engineering and Technology 39.

Thirumalaiah, K., and Deo, M. C., (2000), “Hydrological forecasting using neural networks”, Journal of Hydrologic Engineering 5(2), 180–189.

TrichakisIoannis C., Ioannis K., Nikolos and Karatzas. G. P. (2010) “Artificial neural network based modeling for Karstic groundwater level simulation,” water resources management, DOI: 10.1007/s11269-010-9628-6.

Tsoukalas, L. H., and Uhrig R. E., (1997).“Fuzzy and Neural Approach in Engineering”.New York, John Wiley and Sons, Inc., 87.

Tularam G. A. and Keeler H. P. (2006) “The study of coastal groundwater depth and salinity variation using time series analysis,” journalof Environmental Impact Assessment Review, 26, 633-642.

Virili,F. And Freisleben,B.(2000) “Nonstationary and data preprocessing for neural network predictions of an economic time series”.,IEEE,129-134.

Wang Liying and Zhao Weiguo., (2010). “Forecasting groundwater level based on wavelet network model combined with genetic algorithm”, advanced material research, 113-114, 195-198.

Wang Liying and Zhao Weiguo., (2010). “Forecasting groundwater level based on relevance vector machine”, advanced material research, 121-122, 43-47.

Widrow, B., and M. A. Lehr. (1992). 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. In Neural Networks – Theoretical Foundations and Analysis edited by Clifford Lau, 27-53, IEEE, NY, NY.

Wu,C.L. and Chau,K.W (2010),"Data-driven models for monthly stream flow time series prediction", Elsevier, Engineering Applications of Artificial Intelligence, 23 (2010) 1350–1367.

Xu,L. And Chen,W.J.(2001). "Short term load forecasting techniques using ANN."In: proceedings of the 2001 IEEE International Conference of control Applications,157-160.

Yang Zhongping., LU Wenxi., LONG Yuqiao and LI Ping (2010) "Application of Back-propogation artificial neural network models for prediction of groundwater levels: Case study in western Jilin Province, China," journal of IEEE, 3203-3206.

Yang Zhongping., LU Wenxi., LONG Yuqiao and LI Ping (2010) "Application and comparison of two prediction models for groundwater levels: A case study in western Jilin Province, China," journal of Arid Environments, 73, 487-492.

Yang, C. C., C. S. Tan, and S. O. Prasher. (2000). Artificial neural networks for subsurface drainage and subirrigation systems in Ontario, Canada, Journal of the American Water Resources Association, 36(3): 609-618.

Yang, C. C., S.O. Prasher, R. Lacroix, S. Sreekanth, N. K. Patni, and L. Masse. (1997). Artificial neural network model for subsurface-drained farmlands, Journal of Irrigation and Drainage Engineering – ASCE, 123(4): 285-292.

Zhang, G., Patuwo, B.E., Hu,M.Y., (1998). "Forecasting with artificial neural networks: the state of the art".Int. J. Forecasting 14, 35–62.

Zhang, B. and Govindaraju, R. S. (2000).“ Prediction of watershed runoff using Bayesian concepts and modular neural networks”. *Water Resourcs Research.,* 36(3), 753-762.

LIST OF PUBLICATIONS BASED ON Ph.D. RESEARCH WORK

Papers in Refereed International Journals

- **Aniruddha Gopal Banhatti** and Paresh Chandra Deka **(2012)**.
  Performance Evaluation of Artificial Neural Network Model using Data Preprocessing in Non-Stationary Hydrologic Time Series, CiiT Journal of Artificial Intelligent Systems and Machine Learning, Vol. 4 No. 4, pp 223-228.
- Paresh Chandra Deka,Latifa Haque**,Aniruddha Gopal Banhatti (2012)**
  Discrete Wavelet-ANN Approach in Time series flow forecasting-a case study of Brahmaputra River-Int. Journal of Earth Science and Engg.,vol.5(4),August-2012,pp.673-685.

Papers in International Conferences

- **Aniruddha Gopal Banhatti** and Paresh Chandra Deka**(2012)**.
  Effects of Data Preprocessing on the Prediction Accuracy of Artificial Neural Network Model in Non-Stationary Hydrological Time Series, International Conference on Environmentally Sustainable Urban Ecosystems ‘ENSURE’ 2012 Conference hosted by I.I.T. Guwahati.

BIO – DATA

Name ANIRUDDHA GOPALBANHATTI

Designation Research Scholar

Date of Birth 23rd April, 1956

E- mailanibanister@gmail.com

Contact No. 02065270908

Permanent Address Flat 102, Harmony, Padma Housing Society,

Bibwewadi, PUNE 411037

Educational Qualifications	B.E. Civil,
	M.E. Civil (Hydraulics)
	M.A. French
	Diplome de la langue, French
	Diplome d'Etudes Moderne Francaise
	Diploma in Japanese
	Knows Esperanto Language