International Journal of Science, Environment and Technology, Vol. 2, No 5, 2013, 787 – 804

ISSN 2278-3687 (O)

WEB BASED FORECASTING OF SHORT TERM RESERVOIR INFLOW

A.O. Ibeje[1], J.C. Agunwamba[2] and B.C. Okoro[3]

[1]Department of Civil Engineering, Imo State University, Owerri, Nigeria
[2]Department of Civil Engineering, University of Nigeria, Nsukka, Nigeria
[3]Department of Civil Engineering, Federal University of Technology, Owerri, Nigeria
E-mails: [1]engineeribeje@yahoo.com, [2]nwambaagu@yahoo.com, [3]bcokoro1@yahoo.com

Abstract: In the study, Artificial Neural Network (ANN) model is applied to forecast the daily inflow into Dadin-Kowa Reservoir (River Gongola) in Northern Nigeria. In the study, the 1991-2001 records of forecasted rainfall amounts are solicited as predictors and daily reservoir inflow as predicted targets for Multilayer Perception Artificial Neural Networks (MLP-ANNs) model. The neural network models were trained using the following structure: input node = 1, Hidden Nodes = 7, Training Epochs = 1000.The developed model was integrated into a practical web-based forecasting software for daily inflow predictions of the Dadin-kowa reservoir which is operated by the Upper Benue Development Authority in Gombe State, Nigeria. The software is composed of three main function modules, namely internet connection, download of forecasted rainfall and inflow forecasting modules, respectively.

Keywords: Artificial Neural Network; Model; Daily Reservoir Inflow; Forecasting.

I. INTRODUCTION

The internet, a global network of computer networks, has become one of the world's largest and most promising information resources. The web, an internet service capable of accessing complex information in the form of graphics, styled-text, and even sound and video, has allowed people to increase their sharing and use of information. The use of Web-Based Systems (WBS) has grown dramatically from obscurity in the 1980s to become a common place in businesses, universities and governments; where they are now used for many diverse applications. The web-based environment is emerging as a very important development and delivery platform for real time flood forecasting system and watershed management decision making (Miller et al., 2004). However, hydrological modeling which includes reservoir inflow forecasting, tends to remain in the domain of the model developer and to be applied within a consulting framework. The models are inaccessible to decision makers who are not

*Received August 12, 2013 * Published October 2, 2013 * www.ijset.net*

specialist modelers (Taylor et al., 1998) and appear to be designed for experts and professionals or use as in-house tools (Parson, 1999).

It is evident from literatures (Bundock and Raper, 1991; Fankowski et al., 1997; Van Deussen and Kwadijk, 1993) that coupling hydrological model with the web is difficult. Most hydrological softwares often disconnect intermittently from the web (Goodchild, 1992; Raper and livingstone, 1993). This has often caused abrupt cut in communication when the software wants to download data from a website (Collier, 1994 and Haggett, 1998). Thus, web-based hydrological modelling has become hindered. It is therefore very necessary in this study to develop web-based reservoir inflow software that can couple very effectively with the web.

Dadin-Kowa dam was commissioned in 1988 for irrigation, domestic water supply and flood control (Ibeje et al., 2012). Reservoir inflow forecasting is faced with a huge task of timely dissemination of forecasts to users. In moments of flood warning, forecasts of reservoir inflow are not often received by the public in real time. This has very often led to loss of lives and property. In periods of droughts, farmers may not be informed of shortage of irrigation water which often resulted in wilting of crops. The most damaging aspect of lack of timely forecast of reservoir inflow is the loss of huge revenue by the industries that rely on hydropower energy. In other to tackle the problems, this research is set out to develop web-based forecasting software which will enable all water users to assess the inflow forecast in the internet. This would go a long way to ameliorate problems resulting from late information of reservoir inflow predictions.

The objectives of the research are: to develop rainfall-inflow model using Artificial Neural Network (ANN) and to couple the software with Global Weather Forecasting System for daily download of rainfall forecasts that will be used for the daily forecasts of the Dadin-Kowa Reservoir inflow.

The Study Area

The study area is the Dadin-Kowa reservoir. The Dadin Kowa Dam is in Yamaltu local government area of Gombe State in the north east of Nigeria. Dadin-kowa town is located between latitudes 10 to 10^o 20^o N and longitudes 11^{o}01^{o}E and 11^{o}19^{o}E. The dam is located about 35 kilometers to the east of Gombe town, and provides drinking water for the town. The dam was completed by the federal government in 1984, with the goal of providing irrigation and electricity for the planned Gongola sugar plantation project. The water supply project was built at a cost of about ₦8.2billion by CGC Nigeria, a Chinese company. In 2010 it was providing about 30,000 cubic meters daily, treated at a plant three kilometers from the dam before being piped to storage reservoirs in Gombe (Ogbu, 2010). In August 2001 the federal government announced that it would spend $32 million to complete the Dadin Kowa Dam power generation facilities (Nwezeh, 2001). In March 2009 ₦7 billion was allocated to complete the hydro-electrical generation component of the dam, and another ₦500 million to complete the canal, which would irrigate 6,600 hectares of farmland (Timawus, 2010). The climate of Dadin-kowa is characterized by a dry season of eight months, alternating with a four months rainy season. As in other part of Nigerian savanna, the precipitation distribution is mainly triggered by a seasonal shift of the Inter-Tropical Convergence Zone (ITCZ). For the years 1977 to 1995, the mean annual precipitation is 835mm and the mean annual temperature is about 26°C, whereas relative humidity for the area has same pattern 94% in August and dropping to less than 10% during the harmattan period (Yamaltu L.G.A., 1999). The relief of the town ranges between 650m in the western part to 370m in the eastern parts.

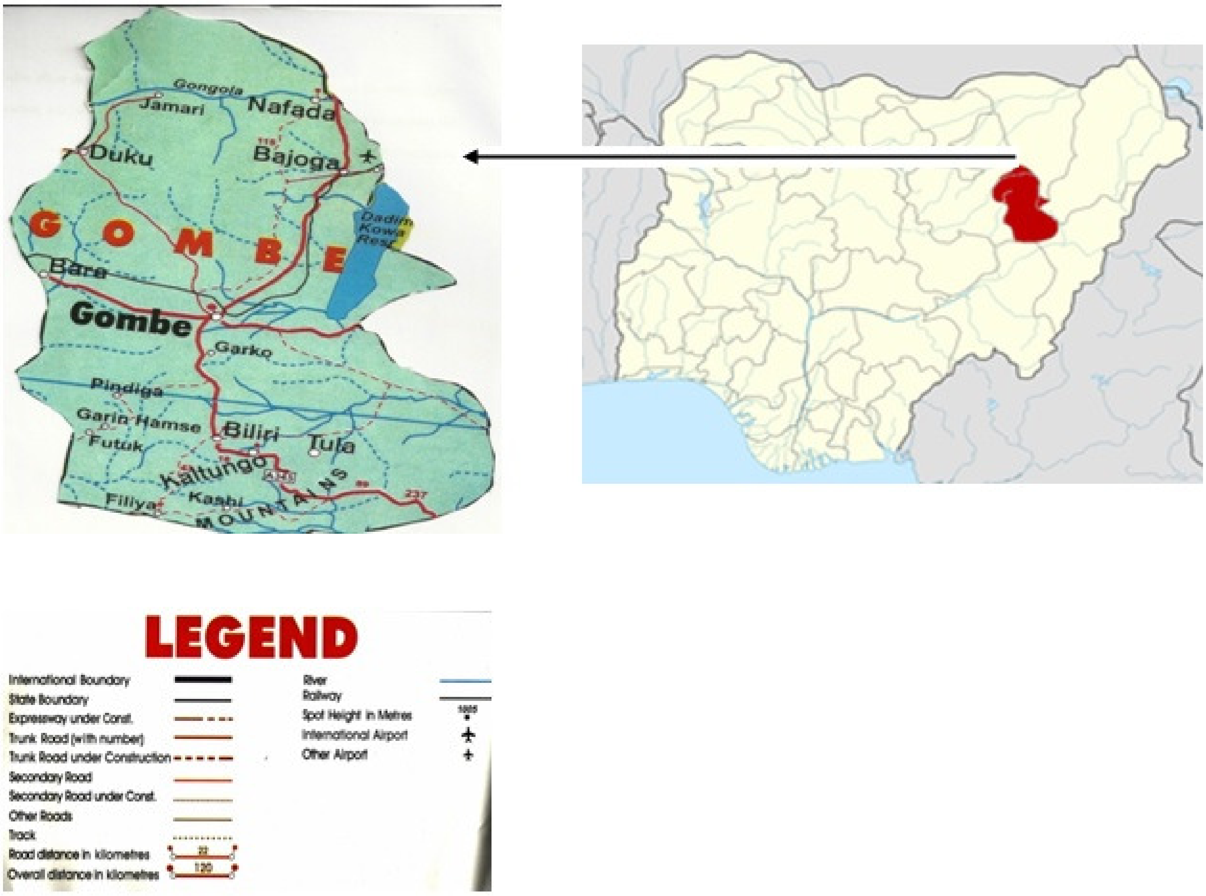

Figure 1: The Location of Dandi Kowa

Reservoir

Dadin-Kowa Dam is a multipurpose dam which impounds a large reservoir of water from Gongola River. It has a storage capacity of 1.77 billion cubic meters for irrigation to 950km^2 area of farmland. (Ibeje et al., 2012). Its flood spillway has a discharge capacity of 1.1110m^3/s.

Dadin-Kowa Reservoir Inflow Characteristics

Figures 2 and 3 show the variation in the inflow to Dadin-kowa Reservoir. The inflow increased gradually after the commissioning of the dam in 1988 until there was a decline in the inter-annual inflow in 1997. The effect of this is the inability to fill the reservoir due to siltation. The flow also exhibited a noisy pattern. The annual inflow showed that the reservoir inflow increased from the month of May to August, after which it declined to the month of September. The month of August is notably the wettest month. Other months in the year never had any reasonable records of inflow over the years. This is a clear demonstration of the climatic characteristics of the reservoir catchment area. There were usually no rainfalls during those months of no inflows.

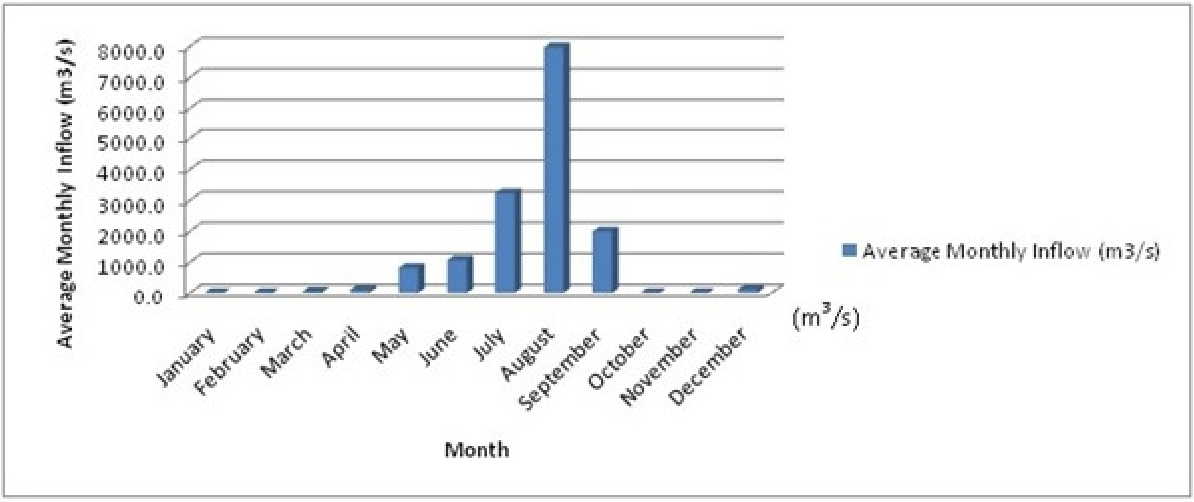

Figure 2: Annual Inflow of Dadin-Kowa Reservoir

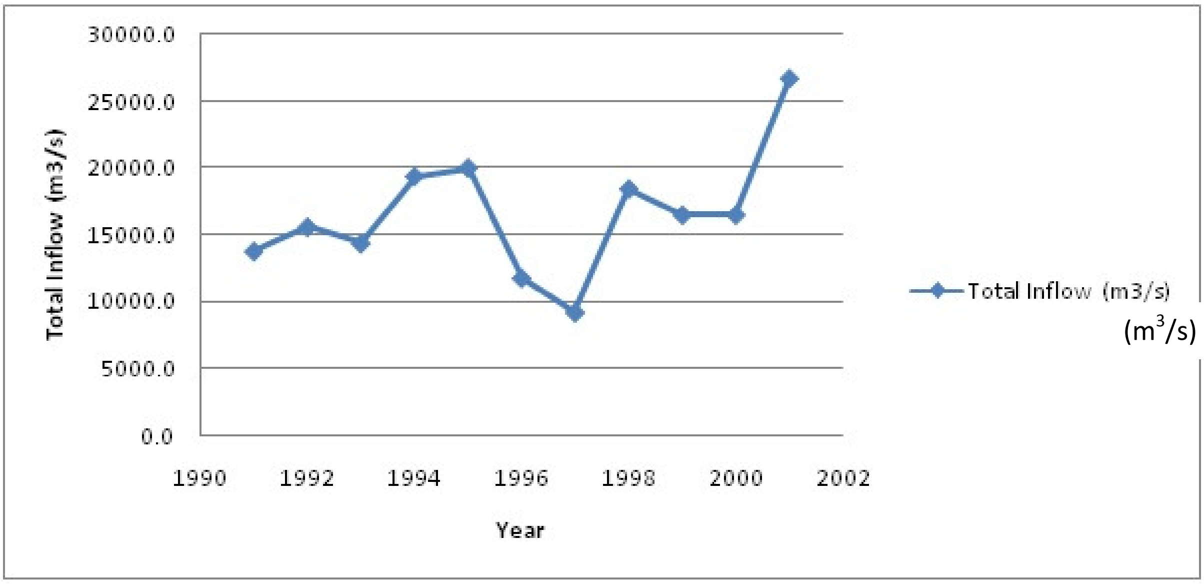

Figure 3: Seasonal Inflow of Dadin-Kowa Reservoir

II. METHODOLOGY

The number of predictors and predicands specified the number of neurons in the input and output layers respectively. An experiment with trial-and-error measure, recommended as the best strategy by Shamseldin (1997) is used to determine the number of neurons in the hidden layer. In general, the architecture of multi-layer MLP-ANN can have many layers where a layer represents a set of parallel processing units (nodes). The three-layer MLP-ANN used in this study contains only one intermediate (hidden) layer. MLP-ANN can have more than one hidden layer; however theoretical works have shown that a single hidden layer is sufficient for ANNs to approximate any complex nonlinear function (Cybenko, 1989; Horinik et al., 1989). Indeed many experimental results seem to confirm that one hidden layer may be enough for most forecasting problems (Zhang et al., 1988). Therefore, in the study, one hidden layer is used. It is the hidden layer nodes that allow the network to detect and capture the relevant pattern(s) in the data, and to perform complex nonlinear mapping between the input and the output variables. The sole role of the input layer of nodes is to relay the external inputs to the neurons of the hidden layer. Hence, the number of input nodes corresponds to the number of input variables. The outputs of the hidden layer are passed to the last (or output) layer which provides the final output of the network. The network ability to learn from examples and to generalize depends on the number of hidden nodes. A too small network (i.e. with every few hidden nodes) will have difficulty learning the data, while a too complex network tends to overfit the training samples and thus has a poor generalization capability. Therefore, in this research, the trial-and-error method commonly used for network design was used.

Performance Assessment of Rainfall-Inflow Model

A number of error measures (Dawson et al; 2007; Legatees and McCabe, 1999) have been developed to assess the goodness of fit performance of hydrological forecasting models but no standard has been specified since each measure can just assess one or two aspects of the runoff characteristic. Three commonly used error measures, therefore, were employed in this study to make the evaluation of the forecasts. They are the Mean Absolute Error (MAE), the Mean Squared Relative Error (MSRE) and the Coefficient of Determination (R^2), respectively defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} /Q_i - \hat{Q}_i/}{n} \quad (1)$$

$$MSRE = \frac{\sum_{i=1}^{n} \frac{(Q_i - \hat{Q}_i)^2}{Q_i^2}}{n} \quad (2)$$

$$R^2 = \left[\frac{\sum_{i=1}^{n}(Q_i - \bar{Q})(\hat{Q}_i - \tilde{Q})}{\sqrt{\sum_{i=1}^{n}(Q_i - \bar{Q})^2(\hat{Q}_i - \tilde{Q})^2}}\right]^2 \quad (3)$$

Where Q_i is the observed discharge, $\hat{Q}_i$ is the simulated discharge, $\bar{Q}$ is the mean of the observed discharges, $\tilde{Q}$ is the mean of the simulated discharges and n is the length of the observed/simulated series.The MAE (Nash and Sutcliffe., 1970), which ranged from 0 to + ∞, was used to measure how close forecasts were to the eventual outcomes. Theoretically, a coefficient of zero (MAE = 0) meant the best model with a perfect performance. The MSRE, which ranged from 0 to + ∞, could provide a balanced evaluation of the goodness of fit of the model as it was more sensitive to the larger relative errors caused by the low value and the best coefficient would be zero (MSRE = 0). The $R^{2,}$ which ranged from 0 to 1, was a statistical measure of how well the regression line close to the observed data and coefficient of one (R^2=1) indicated that the regression line perfectly fitted the observed data.

Design of Reservoir Inflow Forecasting Software

WBS was designed as a component-based software system. The component of GUI Controller accepted all HTTP requests and translated the requests to events. It passed these events to other components for processing and re-directed to correct resulting pages. The component of User Account Management (UAM) handled user account. It implemented the WBS user management functionality. Administrator of WBS used this functionality to manage the user registration and user information. The core component of the system is the

Forecasting Engine. It was designed as an independent deployable component that can be assembled with other system. Login and Session Management provided the authentication for end users and stored session related data. Database Access Layer linked with the dataBase (OB) and Knowledge Base (KB) to store and retrieve the data. Event Logger was used by all other component to record significant events into log files. XML utilities were used to transfer and standardize the data from backend data sources to WBS database.

Graphical User Interface (GUI) Design

Within the scope of the project, a Graphical User Interface (GUI) was designed based on the system functional requirements and the workflow. The GUI of the system was designed with the Web enabled capability by using Microsoft FrontPage 2000. It covered in two different portions: access control and users menu.

Forecasting Engine Design

WIFS forecasting engine was designed with four main modules and four supporting modules. The four main modules are user forecast, forecast configuration, analyzer and data extractor. The four supporting modules are data access layer, knowledge base, database and component interface API. The main modules received requirements from EJB controller through the component interface API, then extracted and used data and knowledge from the supporting modules. Their decisions or results were in turn used to update the database and the knowledge base through the data access layer.

Software Workflow Design

The workflow design defined and described how WBS system works from the user point of view. Workflow diagrams were used to represent system scenario and dynamic behavior.

Web Tier/EJB Controller Design

The design of Web tier and EJB controller defined the way to achieve the system functionality in the JSP (Java Server Page) and EJB (Enterprise Java Bean). The Model-View-Controller (MVC) application architecture concept was adopted. MVC helped in the process of breading an application up into logic components that can be structured more easily. JSP technology was chosen to provide a method of developing servlets. Along with all benefit, servlets offer, JSP offers the capability to rapidly develop servlets where content and display logic are separated, and to reuse code through a component-based architecture. Separate EJB controller was implemented for each component. They could be plugged in and played in application server platform without affecting other components. This also facilitated the maintenance of WBS system.

Database Schema Design

Both logic schema and physical schema models for WBS core database were defined. The logical schema model was used as the key input of the physical schema model, which in turn, served as the main input to the physical implementation of the WBS core database.

Database Access Layer API Design

The Database Access Layer (DAL) was used by a number of modules that was designed to perform different functional tasks. These tasks included the execution of user forecasting, configuration of target, configuration of methods, analysis of forecasting results, presentation of data to the web-tier, management of user access control, and the import data utility.

Materials for Developing Web-based Forecasting System (WBFS)

Support tools and resources for building the WBFS comprised of data-collection hardware, computer hardware and software. Reliability and cost were major factors in selecting both hardware and software. To reduce costs, free software packages available on the internet were used wherever possible.

Data-collection Hardware: Data capture and input, generally the most time-consuming task in building any WBS, have two main requirements (Worboys, et. al. 1993). First, one provided both the physical devices for capturing data external to the system and a method for inputting the captured data into a database. The devices that were used in the WBFS were automatic loggers, wireless modems, and NT servers. The field data were collected automatically in real time and transmitted via wireless modem to an NT server for storage. It was downloaded in nearly real time by FTP and stored in a database, which is immediately accessible to the WBS model for further analysis. The field data is daily rainfall forecasts. A Java-based subsystem is used in the WBFS.

Software Component: A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. It could be independently deployed and run on a predetermined deployment infrastructure or framework.The main advantages of software component are their scalability and reusability. The software component technology was applied to the WBS architecture system design, Web tier/EJB controller design, forecasting engine design and database tier design.

J2EE Technology: Java 2 Enterprise Edition (J2EE) was developed by Sun Microsystems, Inc in 1995. J2EE consists of multitier applications by replacing the client application with a Web browser and HTML pages powered by servlet/JavaServer Pages (JSP) technology. WBS was designed in J2EE three-tier architecture. JSP was adopted to create dynamic pages for implementing presentation in Graphical User Interface (GUI). Each page was implemented with JSP page that talked to JavaBean class to retrieve data. JavaBean was also a part of JEE technology. It is a reusable, interoperable software component that can be visually manipulated with builder tool. Session beans and entity beans were applied in the WBS Web tier/EJB controller design.

Weblogic Application Server: WebLogic application server allows a user to quickly develop and deploy reliable, secure, scalable and manageable applications. It manages system-level details so that the user can concentrate on business logic and presentation.

Computer Hardware and Application Packages

Web-Server Computer: A dell PowerEdge computer, with a 1000MHz Intel Pentium M processor, 400-Mb memory, and an 500 Gb hard disk, was chosen for the Web server.

Database-server Computer*:* The database-server computer is similar to that of the Web server computer. A database server allows a client to pass Structured Query Language (SQL) request messages to the database, which resides on the same computer, and to return results.

Software: All software packages and data sets used were obtained free from the Internet or a government agency.

Operating System: Windows NT 4.0 was selected for the software, over Unix. Although not the best technical choice, because it limits the choice of Web servers, it is sufficient to run the system and is cheaper.

Web Server: A Web server supports the Internet protocols, including HTTP. For the software, Microsoft's Internet Information Server (IIS) was chosen. Although other Web servers were considered, IIS was selected mainly because it is part of the Windows NT option pack and can be downloaded from the Internet. For the Java-based application, Servertec Internet Server 1.10.3 was used. It is a small, fast, scalable, and easy-to-administer platform-independent application and Web server.

Database System: The database system is clearly a fundamental component. Some database systems are large and multi-featured, like Oracle and Sybase, but are expensive. Others are small and free, like mSQL and MySQL mSQL were selected, primarily because the storage requirement was not large. An operational system would require a more powerful database system.

Java-servlet Engine: Because the Web server chosen did not support the Java servlet API, a Java-servlet engine was needed. Although primarily a Web server, the choice was the Servertec Internet Server, which can also be used as a servlet engine for any application or

Web-Server supporting Apache Modules, ISAPI, NSAPI, CGI or Java It runs on any platform supporting the Java Runtime Environment (IRE) version 1.1.x or later, and supports all industry-standard platforms. A servlet can be thought of as a Java applet that runs on the server side.

Development Tools: Development languages that were used include Java, Pen, and HTML, with various Web browsers. The Java Development Kit 1.1.8 Java compiler was used. Java has two important APIs, the applet API and the servlet API. Both were used for model calculation and drawing charts and servlets for data querying.

Structured Query Language: Structured Query Language (SQL) was used to retrieve, create, update, and delete tuples in a relational database. mSQL was used with mSQL-JDBC, a database-access API for the mSQL database engine that conforms to the Java Database Connectivity (JDBC) API. mSQL and the mSQL-JDBC are freely available for non-commercial use. The JDBC API supported both two-tier and three-tier models for database access.

Java-based Applications: Because users often experience difficulty transferring their existing knowledge to the realm of data handling (Davis and Medycki-Scott. 1994), careful attention must be paid to the user interface (Chopra, 1996; Peng, 1997; Wang, 1997). The software Java-based interface used the AWT7° library (Krutsch et al, 2001; Rodngues, 2001),

Procedure for Implementation of Web-based Forecasting System

System Requirements: Functional requirements for web-based modeling software include the ability to invoke remote services, share information, and execute functions across heterogeneous computing environments. More specifically, the requirements include;

- Real-time data acquisition and analysis
- User-side operation with a Web browser only
- Performance of under a few seconds per request
- Low-cost, particularly for the user

Technical support requirements include hardware, software, an Internet connection, and some development tools. The Internet connection had high-bandwidth and connected to the Internet through a local area network with a firewall. Firewall is a system that protects a local area network from unauthorized Internet access, a firewall server controls all communication passing between the Internet and the local network.

System Architecture: The data collection sub-system, providing data from devices in the field, fed into the dynamic model.The dynamic model communicated with both the DBMS server, shown in the lower left, and the Java user interface. Finally, the user's web browser, ran on the user machine. With limited hardware and software resources, the functionality of the software was moderate yet it provided frequently used hydrological techniques. It is a raster-based system permitting the integration of environmental modelling functions with classical WBS functions such as database maintenance and screen display. WBS functions and modelling functions were incorporated in a single WBS–modelling language for performing both WBS and modelling operations. The following section present some of the factors and reasoning that was considered as the prototype that was designed.

Database Connection: The software used two data-retrieval channels. One downloaded data files for local rendering or analyzing. These files were pre-deployed onto the Web server; an applet simply downloaded them from the appropriate URL using standard HTTP. The second obtained statistical, availability, and other analyses about data. This latter channel was isolated inside the DataManager interface. The software was configured to support three database engines, mSQL, MySQL, and Access. The data managers available are:

- mSQL direct connection — used in current version
- Access ODBC direct connection — ready for connection to Access
- MySOL direct connection — ready for connection to MySOL
- Servlet with mSQL backend — used in current version
- Servlet with Access backend — ready for connection to Access
- Servlet with MySQL backend — ready for connection to MySQL

The first three data managers allowed an applet to manipulate the data directly using a JDBC connection. These represented direct-connection methods a 2-tier approach. The last three data managers, allowing an applet to manipulate data using a special proxy servlet using a three-tier approach, represented servlet connection methods.

Direct Connection: A Java database connection was chosen over the combination of HTML forms, CGI, and scripts because of its superior handling of complex systems. JDBC (Hamilton et al, 1998), a Java API for executing SQL statements, consisted of a set of classes and interfaces written in Java (Horstmann et al., 1984) and provides access to a wide range of relational databases. The JDBC driver category that was employed was implemented using a native-protocol pure Java driver. This type of driver converted a JDBC call directly into the network protocol used by the DBMS, allowing for a direct call from the client machine to the DBMS server. An applet was downloaded from a Web server to the client's machine where it used JDBC to access the server. The applet was restricted by the Web-browser security so it could not access the client's local files and could only make a connection to the originating host. The driver accessed the database in the server by communicating with the JDBC applet driver provided by the database vendor. Two security issues related to JDBC, applets and firewalls, merit consideration. Applets were not allowed to connect to servers other than those from which they came. Of course, if one signed an applet one can access any data and execute any program. Firewalls, and some routers, restrict access by restricting ports; therefore they must be carefully designed.

Servlet Connection: Three-tiered applications consisted of a client supporting the user interface, a Java servlet incorporating the application logic, and the database. Processes can be managed and deployed separately from the GUI and the database. Three-tiered systems were claimed to be more scalable, robust, and flexible (Orfali et al., 1996). They can integrate data from multiple sources.

Database Manager Implementation: The software contained two implementations of the Data Manager interface, BaseDBDataManager and Servlet Data Manager. Servlet Data Manager, developed to take advantage of the three-tier architecture, is in fact just a very simple forwarding mechanism. The user interface encoded information and sent it to the servlet in the middle tier, which decoded it and passed it to the appropriate DataManager. After a database connection had been established, through a JDBC or servlet data manager, the servlet data-manager page was activated.

Java User Interface: The Java GUI comprised three pages. The analysis page provided analysis function including slope, aspect, accumulated flux, overlay, zoom, histogram, and metadata. The time-series (TSS) graph page provided time-series analysis. The data-manager page provided database querying via JDBC or the servlet.

III. RESULTS AND DISCUSSION

The learning rate, α and the momentum coefficient, β were automatically adjusted according to Sudhear (2002) to yield: α= 0.001 and β = 0.85. By assigning different hidden node for the neural networks model; the number of input nodes was varied to identify the best input node required by the neural network. The best R^2 tests were produced with one input node, i.e. when only the last inflow lag was used. By assigning input node to one and changing the number of hidden nodes, the best number of hidden nodes for the networkwas obtained as 7. The analysis to find the adequate training epochs was carried out to obtain a training epoch of

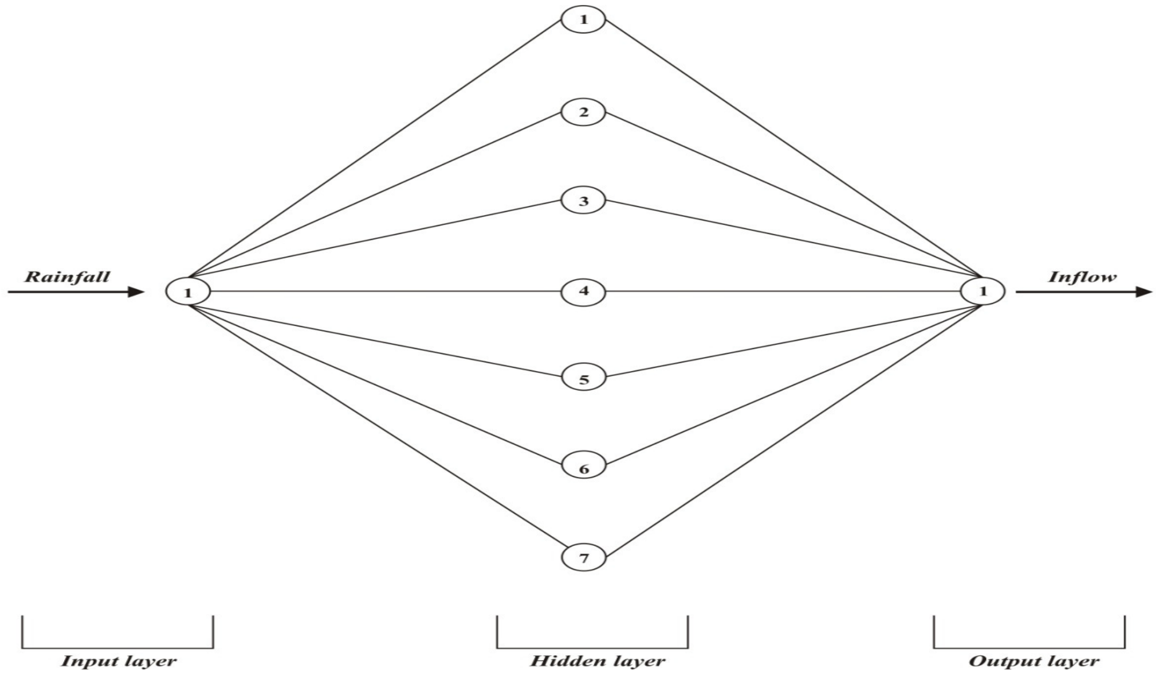

Figure 4: The Architecture of the MLP-BP-ANN model

1000. The neural network models were trained using the following structure: MLP-BP: input node = 1, Hidden Nodes = 7, Training Epochs = 1000.

R^2 tests for lead-time from 1-day up to 6-day of the inflow were calculated over both the training and independent data sets. The results for training data set indicate that MLP-BP gave good R^2 tests up to 4-day, 5-day and 6-day lead-time respectively, where their R^2 test values are about 0.8. The results for independent data set showed that MLP-BP gave good R^2 tests up to 4-day, 5-day and 6-day ahead, respectively.

Results of Model Performance Assessment

From the results (Table 1) shown in the graphs and calculations, a satisfactory forecasting can be obtained in the study, since the r^2 is sufficiently high and close to 1, and the MSRE is adequately low and approximates to 0. The MAE of calibration and validation are far less than the relevant mean value of the observed data. The high scores of r^2 indicate that all the models present the "best" performance according to the standard given by Dawson et al. (2007). The outcome of the modelling implies that the training procedures are successful without "overtraining" or "local minimum" and the proposed models have powerful generalization abilities for out-of-sample forecasting. The architecture (Figure 5) of the best MLP-BP-ANN model for forecasting reservoir inflow was composed of one input layer with one input node, one hidden layer with seven nodes and one output layer with one output variable. The correlation coefficient (R), the mean absolute error (MAE), the mean squared relative error (MSRE), for the calibration, the validation and the verification data set, are given in Table 1. The notation (Q_R/ MLP-BP-ANN: 1-7-1/0.9957) means that the best architecture of the specific MLP-BP-ANN model is composed of one input layer with one input node, one hidden layer with seven nodes and one output layer with one output variables, with value of correlation coefficient equals to 0.9957. According to the results of Table 1, it could be seen that the difference in the R, MAE and MSRE obtained using the test data was not markedly different from that obtained using the training data, thus meaning that there was no overfitting. Also, the results of Table 1 show a good performance of the chosen MLP-BP-ANN model for forecasting the Dadin-Kowa reservoir daily inflow.

Table 1: Results for Model Error Measures for the Calibration, the Validation and Verification Data sets

Q_R/ MLP-BP-ANN: 1-7-1/0.9957			
Data	**R**	**MAE×10^{-5}**	**MSRE×10^{-7}**
Calibration (1991-1998)	0.9957	0.7156	1.4984
Validation (2010-2011)	0.9946	1.1046	1.4035
Verification 1999-2001	0.9688	1.1478	1.1478

Reservoir Inflow Forecasting Software

On the basis of the rainfall-inflow models developed, an integrated web-based daily reservoir inflow forecasting software (WBDIFS) was implemented within java technology framework. It was conformed to the J2EE criterion and a number of Java techniques, such as Applet, Servlet, Swing, JDBC and JSP, were used during the implementation. By the aid of this system, a web-based forecasting of daily reservoir inflow was automatically accomplished in time every day after the (Quantitative Precipitation Forecast) QPF was released. This software was composed of three main function modules, namely internet connection, download of forecasted rainfall and inflow forecasting modules, respectively.

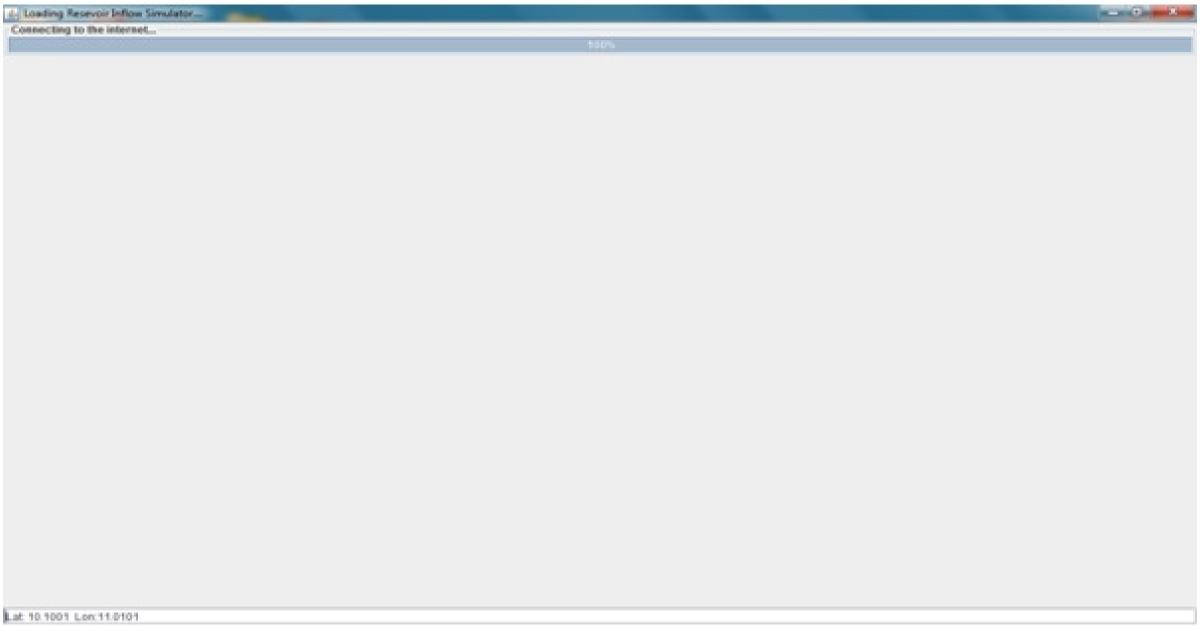

Figure 5: The Internet Connection Module

The inflow forecasting software is the final result of this thesis, the software can make forecasts of the daily inflow into the Dadin-kowa reservoir as well as other reservoir as well as other reservoir located in any semi-arid region anywhere in the world. The only thing that constantly has to be updated is the database containing the forecasted rainfall. The software has an interface which is easy to understand and use. The interface can be seen in Figure 5, 6 and 7.

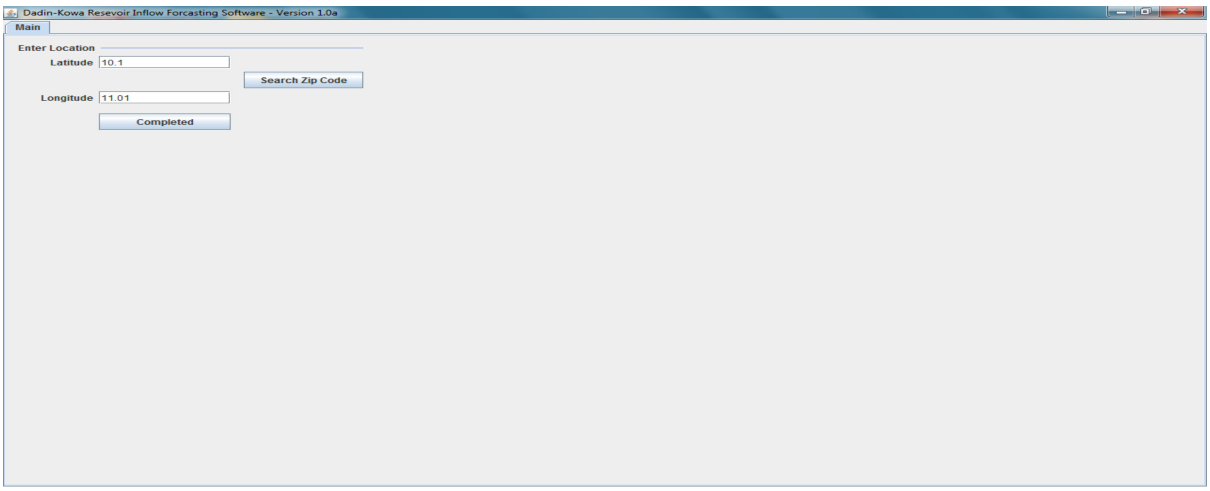

Figure 6: The Forecasted Rainfall (QPF) Download Module

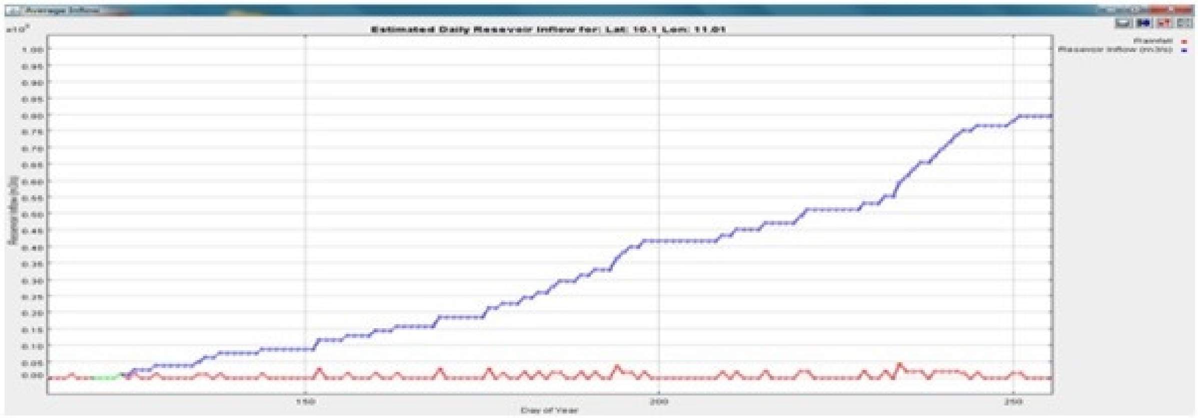

Figure 7: Inflow Forecasting Module

By selecting the week for which the simulation should account for, selecting the day for which it should be run, a simulation will be presented together with the forecasted inflow values for next nine weeks (Figure 5). The inflow forecasting software can be operated by filling in the geometrical coordinates of the location for which the inflow forecast is being made (Figure 6). Then, the inflow forecast would be shown as soon as the "completed button" is clicked (Figure 7).

IV. CONCLUSION

In this work, Multilayer Perceptron Back Propagation Artificial Neural Network (MLP-BP-ANN) models were developed for forecasting daily inflow values into Dadin-Kowa reservoir. The Artificial Neural Network approach becomes more explicit and can be adopted for any reservoir daily inflow forecasting. The experimental results indicate that these models can make inflow forecasts with satisfactory goodness of fit. Integrated web-based daily reservoir inflow forecasting software, implemented within java technology framework, was developed and was successfully used for daily inflow predictions of Dadin-kowa reservoir which is operated by the Upper Benue River Basin Development Authority in Nigeria.

REFERENCES

[1[Bundock, M. and Raper, J. (1991). Development of a Generic Spatial Language Interface for GIS. In book-Geographical Information, Handling-Research and Applications. In Paul Mather (Ed.) Chichester: John Wiley and Sons. Pp 113-143.

[2] Chopra, P.N. (1996). Geological Mapping on the Web. The Australian Geological Survey Organisation Introduces On-line Customizable Maps. GIS USER, 19, pp 35-39.

[3] Collier, C. G. (1994). Objective Methods of Forecasting Precipitation Using Weather Radar Data. In Almeda-Teixeira, M.E., Fantechi, R., Moore, R., Silva, V.M. (Ed). Advances in Radar Technology. European Commission Report EUR 14334 EN

[4] Cybenko, G. (1989). Approximation by Superposition of a Sigmoid Function. Math. Control Signals.. 2, 303-314.

[5] Davis, M.K. and Maidment, D. (2000). Object Oriented Modelling of Rivers and Watershed in Geographical Information System. CRWR. Online Report No 2007.

[6] Dawson, C.W., Abrahart R.J, and See, L.M. (2007). Hydrotest : A web-based Toolbox of Evaluation Metrics for the Standardized Assessment of Hydrological Forecasts, Environ Modell, Soft Ware 22, 1034-1052.

[7] Fankowski, J. M, Goodrich, D.C., Woolhiser, D. A. and Sorooshian, S. (1997). Impact of Small-scale Rainfall Variability on Runoff Simulation. Journal of Hydrology. 173, 309-326.

[8] Goodchild, M. (1992). Integrating GLS and spatial data Analysis: Problems and possibilities International Journal of Geographical Information Systems. London: Taylor and Francis (6) pp 407-432.

[9] Hamilton, G., Fisher, M. and Cattel, R. (1998). JDBC Database Access with Java:A Tutorial and Annoted Reference. Reading, MA: Addison Wesley. Pp. 154-160.

[10] Horstman, D., Young. P. and Taman, N. (1984). Recursive Estimation and Time-series Analysis. Springer-verlag; New York. Pp. 38.

[11] Ibeje, A.O., Agunwamba, J. C. and Okoro, B.C. (2011). Modelling of Water Allocation in Semi-arid Regions. (In press)

[12] Krutsch, K.F., Cargo, D.S. and Howlett, V. (2001). Professional Java Custom UI Components. Wrox Press Inc. Chicago, IL. Pp. 30.

[13] Legatees, D.R. and McCabe, G.J. (1999).Evaluating the Use of “Goodness of Fit” Measures in Hydrologic and Hydroclimatic Model Validation. Water Resour.Res., 35, 233-241.

[14] Miller, R. C., Guertin, D. P. and Hailman, P. (2004). Information Technology in Watershed Management Decision Making. J. Am. Water Resour. Assoc. 40(2). 347-357.

[15] Nash. J.E and Sutcliffe, J.V. (1970). River Forecasting through Conceptual Models Part 1-A Discussion of principles. Hydrol., 10, 282-290

[16] Nwezeh, K. (2001). "FG to Spend $32m On Gombe Dam Project". This Day. Retrieved 2010-05-23.

[17] Ogbu, M. (2010). "Gombe Rural Communities Enjoy Potable Water -Isa". Daily Independent. Retrieved 2010-05-23.

[18] Orfali, R., Hakey, D. and Edwards, J. (1996). The Essential Client-serve Survival Guide. Canada: John Wiley and Sons, Canada. Pp. 72.

[19] Parson, S. (1999). Development of an Internet Watershed Education Tool (Internet) for the Spring Greek Watershed of Central Pennsylvania. A PhD Thesis in Agricultural and Biological Engineering. The Pennsylvania State University.

[20] Peng, Z. P. (1997). An Assessment of the Development of Internet GIS. 1997 ESR, International User Conference. Eric Inc. [On-Line]. Available: http://www.gisdevelopment.net/aars/2000ts7/gdi012.shtm .(Accessed April, 2010).

[21] Rodngness, l. H. (2001). Building Imaging Application with Java Technology: Using AWT Imaging, Java 2D and Java Advanced Imaging (JAI). Boston: Addison-Wesley Longman. Pp. 58-61.

[22] Shamseldin, A.Y. (1997). Application of A Neural Networks Technique to Rainfall-Runoff Modelling. J. Hydrol., 199, pp 272-294

[23] Sudheer, K. B., Gosain, A.K. and Ramasastri, K.S. (2002). A Data-driven Algorithm for Constructing Artificial Neural Network Rainfall-runoff Models. Hydrol. Process., 16, 1325-1330.

[24] Taylor, K., Cameron, M. and Haines, J. (1998). An Integrated Information System on the Web for Catchment Management ACM GIS 98, November, 1998, Washington D.C.

[25] Timawus, M. (2010). "Jonathan Needs More Than Good Luck". Daily Trust. Retrieved 2010-05-23.

[26] Van Deussen, W.P. and Kwadijk, J. C. (1993). RHINE FLOW: An Intergraded GIS Water Balance Model for the River Rhine in Application of Geographic Information, System in Hydro GIS 1993. In Kavar, K. and Nachtnebel, H.P. (Eds) IAHS publication No 211.

[27] Wang, L. (1997). Network-Oriented GIS: Integrating GIS with the Internet. URISA 1997. Annual Conference Proceedings, Toronto, Canada, July 20-24. 1997. CD-Rom

[28] Worboys, M.F., Mason, K.T. and Dawson, B.R.P. (1993). The Object-based Paradigm for Geographical Database System: Modelling Design and Implementation Issues. Georapgical Information Handling-research and Application. Paul Matter (Ed) Chichester; Wiley and Sons publishing company. Pp 91-102.

[29] Yamaltu Local Government Area. (1999). Potrait of Dadin-Kowa L.G.A, Gombe State Nigeria.

[30] Zhang G., Patuwo, B.E. and Hu, M.Y. (1998). Forecasting with Artificial neural Networks: the State of the Art. Int. J. Forecasting 14, 35-62.