

WAFARi: A new modelling system for Seasonal Streamflow Forecasting service of the Bureau of Meteorology, Australia

D. Shin^a, A. Schepen^b, T. Peatey^b, S. Zhou^c, A. MacDonald^a, T. Chia^a, J. Perkins^b and N. Plummer^c

^a Bureau of Meteorology, Canberra, Australian Capital Territory

^b Bureau of Meteorology, Brisbane, Queensland

^c Bureau of Meteorology, Melbourne, Victoria

Email: d.shin@bom.gov.au

Abstract: The Bureau of Meteorology (the Bureau) is responsible for compiling and disseminating comprehensive water resource information for Australia through the Commonwealth Water Act 2007. One of its roles is to provide regular forecasts of streamflows and water availability. To perform this role, the Bureau launched a new Seasonal Streamflow Forecasting (SSF) service in December 2010. The service delivers three-month probabilistic forecasts of total streamflow volumes at single locations or total inflows into a water storage, issued monthly. To ensure the timely and reliable delivery of the SSF service, a new modelling system named WAFARi (Water Availability Forecasts of Australian Rivers) was developed. Using the CSIRO's statistical Bayesian Joint Probability (BJP) system as its kernel, WAFARi is equipped with many tools to support the entire workflow for the streamflow forecasts. These tools range from data management into a central database, to web publication through an operational server. WAFARi stores and manages data in self-descriptive files with fully annotated metadata and provides graphical tools that generate highly tailored visual products for the SSF service. Some of the tools were designed to exploit the parallel computing power of the massive Bureau cluster to accelerate the computationally intensive analysis of BJP. All of these tools are available through the powerful scripting environment of Python, which allows users to inspect intermediate model outcomes at any step and flexibly control their behaviours, either in an interactive command shell or with script files. This paper highlights the major features of the system and explains its application to the SSF service.

Keywords: *streamflow, forecast, model, system, Bureau of Meteorology*

1. INTRODUCTION

The Bureau of Meteorology (the Bureau) is responsible for compiling, disseminating and reporting water resource data for all of Australia through the Commonwealth Water Act 2007. One of its new roles is to provide regular forecasts of streamflows and water availability. As one of the new streamflow forecasting services, the Bureau rolled out the Seasonal Streamflow Forecasting (SSF) service (Plummer *et al.*, 2009) in December 2010. The service issues three-month streamflow volumes at sites or total inflows to water storage reservoirs in the first week of each month (<http://www.bom.gov.au/water/ssf/>).

A requirement for the reliable delivery of SSF service was a robust operational modelling system, and a new modelling system called WAFARi (Water Availability Forecasts of Australian Rivers) was developed to meet this requirement. After undergoing one year of testing for a pilot service, WAFARi was commissioned as the official modelling system to generate SSF products. The purpose of this paper is to introduce this new modelling system and to highlight its major features. It also briefly describes how the system is being operated for the SSF service.

2. BAYESIAN JOINT PROBABILITY

The current operational SSF products are being generated using the Bayesian Joint Probability (BJP) approach. The BJP approach transforms a set of streamflows and their predictors into a multivariate normal distribution and infers the distribution of model parameters using a Bayesian formulation, which is implemented through a Markov chain Monte Carlo (MCMC) sampling method. The Bayesian inference allows the use of data that contains non-concurrent and missing records. Once the parameters are sampled, they are used along with observed data of predictors for predicting probabilistic distributions of future streamflows. The details of the BJP modelling approach to streamflow forecasting are given in Wang *et al.* (2009) and Wang and Robertson (2011).

Applying BJP for the SSF service requires the following four processes conducted in order:

(1) Predictor selection: Compare different combinations of input variables and select the optimal set of variables. For the SSF service, antecedent observed streamflow, rainfall data and a variety of climate indices are examined as potential predictors, (2) Cross-validation: Evaluate forecasting performance by comparing predicted streamflow against observed data for the entire data recording period of each site. In the process, model parameters are re-sampled at each time step, (3) Parameter sampling: Sample parameter values for each forecast update. For the current SSF service, 12 sets of parameter values are sampled for each month's forecast update. (4) Forecast: Predict the probabilistic distribution of three-month aggregated streamflow using the sampled parameter values and observed predictor data.

The first three processes are computationally demanding and require sophisticated data pre- and post-processing. WAFARi was developed as an integrated modelling system, primarily to facilitate all the four processes for the application of BJP within the operational SSF service.

3. WAFARi MODELLING SYSTEM

3.1. Overall System Structure

Figure 1 shows the overall structure of WAFARi system. At the centre of WAFARi, there is a Python interpreter, which integrates all different program modules into a system. Python has been widely adopted for scientific and engineering software development, mainly because of its easy and elegant syntax, and the vast pool of third party packages and tools for numerical programming (Oliphant, 2007; Pérez *et al.*, 2011). Among the packages and tools, Table 1 lists third party libraries used for WAFARi development.

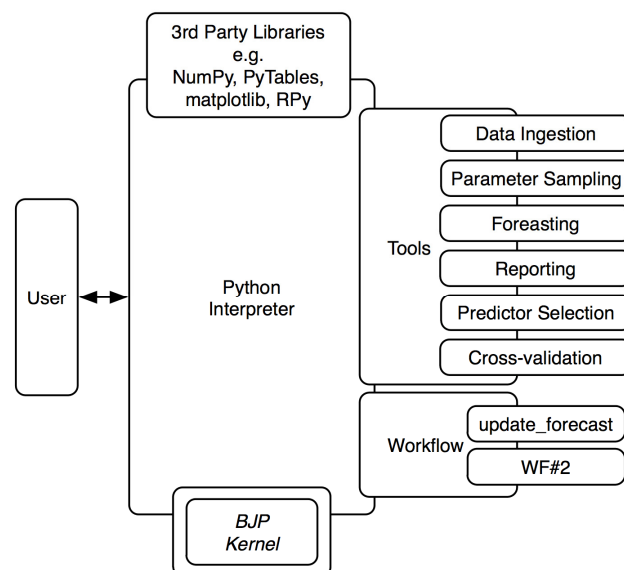


Figure 1 Schematic diagram for the overall system architecture of WAFARi.

Table 1 Major 3rd party libraries and tools used for WAFARi development.

| Name | Description | Web site |
|------------|------------------------------------|---|
| NumPy | Multi-dimensional array structures | http://numpy.scipy.org |
| SciPy | Numerical computation | http://www.scipy.org |
| matplotlib | 2-D plotting | http://matplotlib.sourceforge.net/ |
| PyTables | HDF5 file management | http://www.pytables.org/ |
| Ipython | Enhanced Python command shell | http://ipython.scipy.org |
| Sphinx | Automatic documentation | http://sphinx.pocoo.org |
| Rpy | Interface to R | http://rpy.sourceforge.net |
| Scons | Automatic software building tool | http://www.scons.org |

A BJP model compiled from Fortran programs was wrapped with a Python class and its behaviour is controlled through the method calls of the class. The mixed-language structure makes it possible to leverage both the fast computing speed of a compiled language and the rapid prototyping and testing capability of a dynamic programming language (Ousterhout, 1998).

Overall, WAFARi consists of three major components: core, tools and workflows. The core component is responsible for data exchange between a coupled BJP model and other programs. It also provides common services shared by many tools, such as data input/output and common statistical analysis. Tools are program functions, each of which is designed to conduct a specific task such as parameter sampling or streamflow forecast for users. Tools are classified into several toolboxes depending on their operations. **Table 2** lists the toolboxes currently available in WAFARi. In addition, WAFARi is also equipped with workflow script files, each of which uses multiple tools in a certain sequence to automate routine operations. More details about workflow script files are given later.

Table 2 List of WAFARi toolboxes. To support interactive modelling more effectively, a toolbox is accessible using an alias rather than the full name.

To support a wide range of sophisticated tasks, WAFARi handles a number of configuration and data files, each of which has a standard internal structure. **Table 3** lists the control and data files that WAFARi is currently using. JSON (JavaScript Object Notation) and HDF5 (Hierarchical Data Format ver. 5) were adopted to design configuration files and data files respectively. Both JSON and HDF5 store data on the basis of key-value mapping, rather than its location, which enables the introduction of new variables or attributes to the conventions without braking backward-compatibility. WAFARi is platform-independent, able to run on either Unix-compatible and Windows OS.

| Name | Description | Alias |
|---------------|---------------------------------|-------|
| system | System management tools | sys |
| climate_index | Climate index tools | cx |
| streamflow | Streamflow data tools | sf |
| hydromet | Hydrometeorological data tools | hm |
| parameter | Parameter sampling tools | pm |
| forecast | Streamflow forecast tools | fc |
| xvalidate | Cross-validation tools | xv |
| pselect | Predictor selection tools | ps |
| report | Web publication tools | rp |
| foreign | External system interface tools | fn |
| archive | Backup tools | ar |

Table 3 WAFARi configuration and data files.

3.2. Interactive Simulation and Automation

Users can access all of the WAFARi tools through IPython (Perez and Granger, 2007), an enhanced Python command shell. In the interactive command shell, users can evolve their simulations by typing each command one-by-one to conduct a certain operation. Whenever a command completes an operation, the user can inspect, analyse and verify the results using tools.

| File name | Convention description | Format |
|---------------------|-------------------------------|--------|
| system.json | System configuration | JSON |
| basin.json | Basin configuration | JSON |
| simopts.json | Simulation options | JSON |
| report.json | Web publication configuration | JSON |
| web-content.json | Web site contents | JSON |
| website_config.json | Web site configuration | JSON |
| climate_index.hdf5 | Climate indices | HDF5 |
| hydromet.hdf5 | Hydrometeorological data | HDF5 |
| streamflow.hdf5 | Streamflow data | HDF5 |
| parameter.hdf5 | Sampled parameters | HDF5 |
| forecast.hdf5 | Forecast outcome | HDF5 |
| xvalidate.hdf5 | Cross-validation outcome | HDF5 |
| pselect.hdf5 | Predictor selection outcome | HDF5 |

Through this iterative and incremental exercise, users can approach more effective and efficient solution for a given problem. The support of interactive and incremental simulation has proved useful for exploratory testing on new sites or new methods.

On the other hand, WAFARi is also designed to fully automate a series of operations through scripting. In fact, the entire WAFARi system was implemented as a package, which can be imported to any Python script file. Once the entire system is imported, most components in the system become accessible to the script file. Users can use the same function signatures in a script file as they type in a command shell, so they can apply what they have learnt through interactive exercise in a command shell, when writing a script file to automate their data process. The scripting capability of WAFARi encourages experienced users to try new solutions by themselves, which eventually contributes to the functional expansion of the WAFARi system.

Some of script files were integrated as a component of WAFARi to facilitate routine operations typically conducted for the delivery of SSF products. Once invoked by a single line command, the workflow script files are designed to complete a complex series of operations without any intervention from users. In this way, WAFARi significantly improves the productivity and efficiency of its users and help them spend more time and resources in analyzing forecast products.

3.3. Self-descriptive File Formats

Metadata is the descriptive data about the data content, which assists in correctly interpreting stored data. Without enough metadata, data files are vulnerable to the risk of degenerating into indecipherable puzzles. To prevent this problem, all input and output data files of WAFARi (**Table 3**) were designed to be self-descriptive and platform-independent. The data files are based on HDF5, a standard file format adopted by many scientific and engineering communities. A unique feature of HDF5 is the ability to construct a hierarchical structure within a binary file. Using this feature, a number of heterogeneous variables can be stored to a single binary file in a systematic way. Additional attributes of data, such as data units and filled values, can be attached as metadata. WAFARi uses the object-oriented API of PyTables to access data and metadata in the data files. The interface enables efficient and convenient access to data stored at any position in the internal hierarchy, even when a data file has a complex structure.

3.4. Tailored Data Visualisation

WAFARi uses Matplotlib package (Hunter, 2007) as the primary package for data visualisation. As well as providing many useful plotting routines out-of-the-box, the package provides an object-oriented interface that gives programmers great flexibility in customizing a graph. Using Matplotlib, we have developed many customised plots for the SSF service, which are automatically generated by WAFARi tools. **Figure 2** shows several examples of the plots produced by WAFARi for the SSF service.

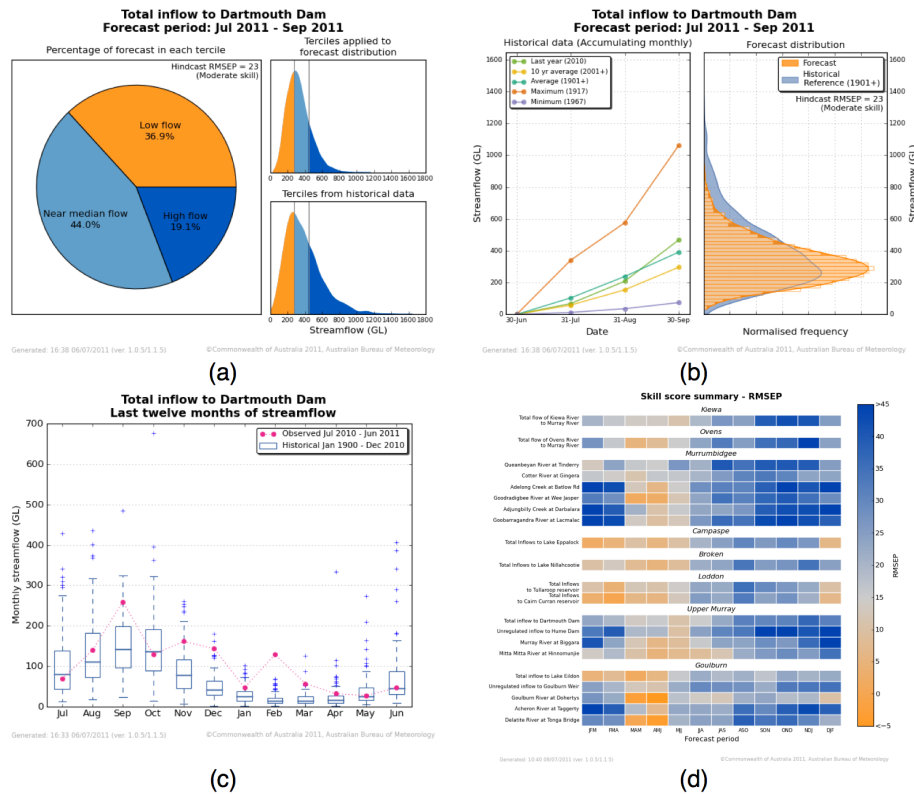


Figure 2 Several SSF graphic products that WAFARi generates. (a) Forecast tercile, (b) Streamflow forecast probability distribution with historical trends, (c) Monthly observed streamflow boxplot and (d) Monthly skill score summary.

3.5. Parallel Computing

Developing a new streamflow forecasting model using the BJP modelling approach is expensive in terms of computing time. The model development and verification tasks of predictor selection and cross-validation can take days to complete when run on a single processor. To overcome the problem, we have implemented parallel computing code that allows these tasks to be performed simultaneously in an existing cluster of the Bureau.

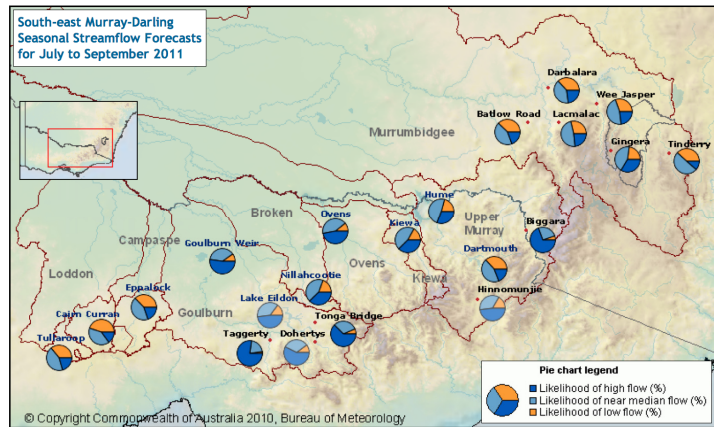
The parallelisation is implemented at two levels. At the first level, access to shared data files is coordinated using a simple file lock utility. It allows users to spawn multiple processes in a cluster environment, without causing any data corruption. At the second level, a master process is used to spawn multiple slave processes that work on different data portions and collect their results into one output file. This solution reduces the total computing time further, particularly on a multi-core computer system. With these two levels of parallelisation, we can have a dramatic decrease of total computing time, particularly for multi-site parameter selection and cross-validation.

4. APPLICATION FOR SEASONAL STREAMFLOW FORECASTING SERVICE

The WAFARi system was tested for one year to support a pilot SSF service, which was accessible only to registered users. During this period, new user demands were identified, but WAFARi could accommodate most of the new demands quickly, mainly due to its flexible and extendible structure. Since the official release last December, WAFARi is being used as the official modelling system to produce seasonal streamflow forecast products and also evaluate model performance for new sites or upgraded algorithms.

Figure 3 shows the locations of sites where WAFARi is currently being applied to produce forecast products. A project is currently under way to extend the service coverage beyond the southern Murray Darling Basin, based on the needs of key stakeholders.

Figure 3 Site locations under SSF service. The maps also show forecast terciles of Jul-Aug-Sep streamflow in 2011. For the details of the terciles, refer to <http://www.bom.gov.au/water/ssf/>.



Every month, a workflow script file is invoked to update the SSF products on the web site. **Figure 4** shows the sequence of major operations and the corresponding tools that are called from the script file. The script file conducts a variety of operations from data ingestion through to forecast generation, and web publication, as required for updating forecast products of all SSF sites.

Figure 5 shows snapshots of several web pages in the web site. The web sites consists of maps, graphs, and text linked together to provide observed data, probabilistic streamflow forecasts, and cross-validation results of each site as well as summary charts for inter-site comparison. In fact, WAFARi also provides tools to automatically generate and publish most of the contents for the web site from several templates, which simplifies the entire publication process of SSF.

5. SUMMARY AND FUTURE WORK

As a leading operational modelling system in the Bureau for SSF service, WAFARi provides a suite of tools to support the application of BJP for the service. The system features interactive simulation, self-descriptive file formats, a variety of visualization tools, multi-level parallelization and powerful scripting capability. It has been demonstrated that the structure of WAFARi is flexible and extensible, allowing it to accommodate new user demands quickly.

The improvement and expansion of the functional capability of the system is driven by consistent stakeholders and user engagement. Some of goals in the near term are:

- Expand its support for the integration between dynamic modelling and statistical modelling results.
- Implement more effective parallelization solution to reduce the entire processing time further.

ACKNOWLEDGMENTS

The WAFARi development is supported by the Water Information Research and Development Alliance (WIRADA) between the Bureau and the CSIRO Water for a Healthy Country Flagship Program. The authors acknowledge the support from Narendra Kumar Tuteja and Richard Laugesen in the Extended Hydrological Prediction Section of the Bureau and Enli Wang and Hongxing Zheng in CSIRO and the BJP development team of Q. J. Wang and David Robertson in CSIRO.

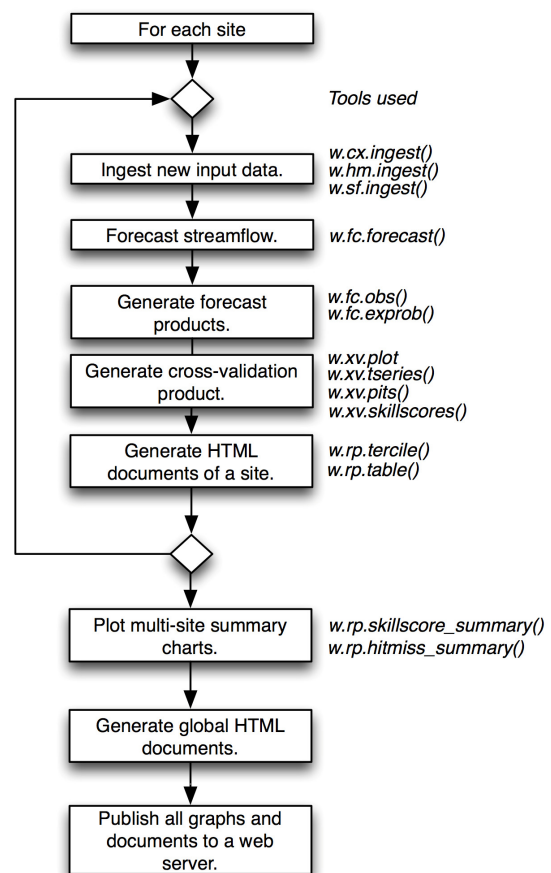


Figure 4 Data flow for SSF update. Major operations and their corresponding WAFARi commands are listed.

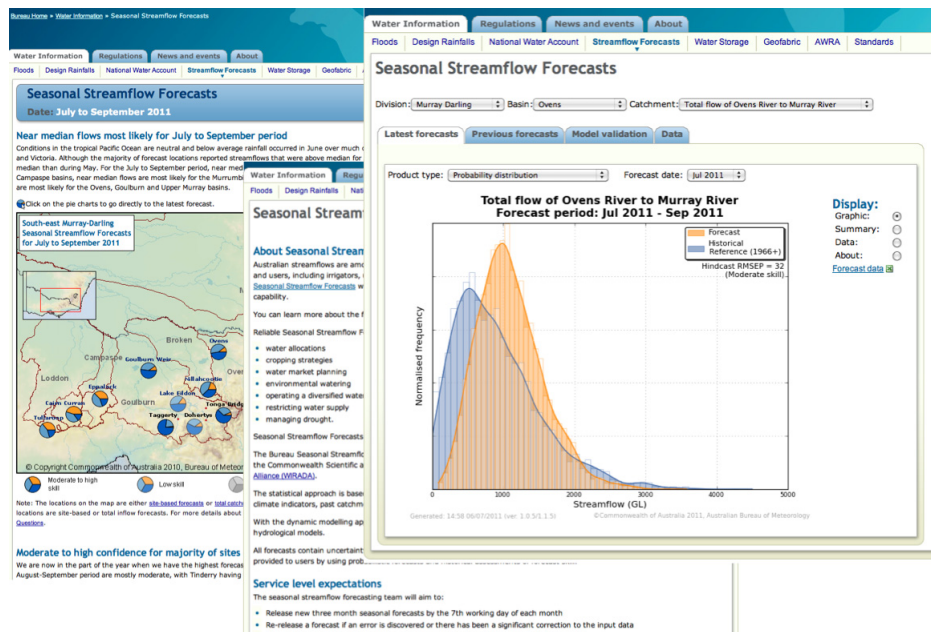


Figure 5 Web pages of SSF service.

REFERENCES

1. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9:90–95.
2. Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9:10–20.
3. Ousterhout, J. K. (1998). Scripting: higher level programming for the 21st century. *IEEE Computer*, 3:23–30.
4. Plummer, N., Tuteja, N.K., Wang, Q.J., Wang, E., Robertson, D., Zhou, S., (2009) A Seasonal Water Availability Prediction Service: Opportunities and Challenges. 18th World IMACS / MODSIM Congress, Cairns, 13-17.
5. Pérez, F. and B. E. Granger (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, 9:21–29.
6. Pérez, F., B. E. Granger, and J. D. Hunter (2011). Python: An ecosystem for scientific computing. *Computing in Science & Engineering*, 13:13–21.
7. Wang, Q. J., D. E. Robertson, and F. H. S. Chiew (2009). A Bayesian joint probability modeling approach for seasonal forecasting of streamflows at multiple sites, *Water Resources Research*, 45, W05407, doi:10.1029/2008WR007355.
8. Wang, Q. J. and D. E. Robertson (2011). Multisite probabilistic forecasting of seasonal flows for streams with zero value occurrences. *Water Resources Research*, 47, W02546, doi:10.1029/2010WR009333.