ACCELERATOR CONTROL DATA MINING WITH WEKA*

W. Fu[†], K. A. Brown, T. D'Ottavio, P. S. Dyer, S. Nemesure
Brookhaven National Laboratory, Upton, USA

Abstract

Accelerator control systems generates and stores many time-series data related to the performance of an accelerator and its support systems. Many of these time series data have detectable change trends and patterns. Being able to timely detect and recognize these data change trends and patterns, analyse and predict the future data changes can provide intelligent ways to improve the controls system with proactive feedback/forward actions. With the help of advanced data mining and machine learning technology, these types of analyses become easier to conduct. As machine learning technology matures with the inclusion of powerful model algorithms, data processing tools, and visualization libraries in different programming languages (e.g. Python, R, Java, etc), it becomes relatively easy for developers to learn and apply machine learning technology to online accelerator control system data. This paper explores time series data analysis and forecasting in the Relativistic Heavy Ion Collider (RHIC) control systems with the Waikato Environment for Knowledge Analysis (**WEKA**) system and its Java data mining APIs.

INTRODUCTION

The Waikato Environment for Knowledge Analysis (WEKA) [1] system provides more than 80 machine learning algorithms and models in the latest version and third party addon packages. This number continues to increase as more models become available over time. Typically, a data mining process with machine learning involves the following steps as shown in Fig. 1. WEKA performs data mining tasks in a similar fashion.

In the RHIC controls system, time-series data is logged and saved in databases or in files using a self-describing data set (SDDS) format. WEKA supports four types of data that includes numeric, nominal, string, and date-time. WEKA also supports data files in ARFF/CVS formats which is able to embed above named basic data types [2]. Due to the data format and type differences, most RHIC controls data cannot be directly loaded into the WEKA system. Proper data conversion and pre-processing are required.

This data conversion process can be simplified by taking advantage of the Collider-Accelerator Department (C-AD)'s existing Database Management Tool (dmt). This application pre-processes the data from a database into the format needed by WEKA. Once the data is imported into WEKA, all applicable data processing models, training models, visualization tools, and other handy tools available in WEKA can be applied to the data.

For both logged and real-time data, we developed a Java program which implemented the WEKA's Java machine learning APIs for data loading, conversion, pre-processing, and model training, and use the program to load and convert logged/live RHIC controls data. The data is then streamed into the WEKA data mining engine providing for easy access to all of WEKA's data mining models.

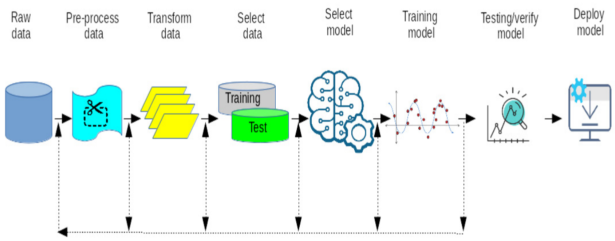


Figure 1: Work flow of data mining with WEKA.

WEKA has a rich set of tools and Java APIs for many kinds of data mining/machine learning tasks. In this paper, we focus on accelerator controls time series data analysis and forecasting.

TIME SERIES DATA FORECASTING

Time series data is a common category of accelerator controls data. A time series data is predictable if:

- The target data series has detectable and recognizable change trends and patterns;
- The target data series has one or more associated data series which have detectable and recognizable change trends and patterns, and these trends and patterns affect the data changes in target data series and make the target data predictable;

Machine learning models can help detect and recognize hidden trends and patterns quickly.

The temporal nature of accelerator time-series data creates the desired need to predict future data trends in order to anticipate and prevent problems and take optimized action ahead of time.

Time series data are typically not suitable for data mining or machine learning techniques which requires each data point to represent an independent observation and independent of data order. Time series analyses [3] use statistical techniques to model a time-dependent series of data. This is usually the better choice for forecasting future data based on historical data. WEKA has a

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy.
[†] Email: fu@bnl.gov

dedicated package for time series forecasts which allows users to develop time-series forecasting models.

"WEKA's time series framework takes a machine learning/data mining approach to modeling time series by transforming the data into a form that standard propositional learning algorithms can process. It does this by removing the temporal ordering of individual input examples by encoding the time dependency via additional input fields. These fields are sometimes referred to as "lagged" variables. Various other fields are also computed automatically to allow the algorithms to ***model trends and seasonality****. After the data has been transformed, any of WEKA's regression algorithms can be applied to learn a model."* [4]

In the RHIC controls system, physicists and controls personnel often need to connect to the RHIC controls network through secure networks as well as via remote connections using NX servers (with a client software named NoMachine). We have been collecting this time series data in databases. Some of the information collected includes user information, user access time periods, user IP address, NX server host connection information, and programs executed. Some problems NX users occasionally experience include connection allocation limitations or difficulties, system slowness, program crashes and discovery of the connecting host with the most available resources. Forecasting analysis of this time series based data may help answer these questions, and in turn, provide the data needed in order to take the right measures to optimize NX server performance.

All data (45K instances in approximately three years) collected are time series with rough patterns. C-AD has four NX server hosts available to users. Once loaded the data into WEKA, the visualization features of WEKA exposes some useful information hidden in the data instantly:

- Distribution of users on the 4 NX server hosts shows some hosts are underutilized. This could provide a useful technique for load-balance automation.
- Distribution of local (on-site) and remote (off-site) NX users. Local usage is 40.66% while remote access is-59.34%. This indicates that, these dedicated machines for remote users are also heavily used locally, Guiding local users to use local machines may make remote users have more usable resource.
- 350 unique users made 45,000 connections from 858 unique IP addresses in about 4 years. This data indicates, there are multiple duplicated concurrent connections per user. This suggests properly user connection control may be needed to avoid user connection resource waste.
- Distribution of monthly usage based on user connections. This gives a clear picture of when the remote connection system are mostly used. And users may avoid the peak time if possible.

The information found above exposes a problematic situation where many individual users have multiple connected NX sessions through one or more NX servers. In fact, further analysis showed that, in one case a single user held 50 concurrent connections. Discoveries such as this can be quickly found and addressed rather than relying on manual or periodic scripts (cron jobs) to perform the analysis.

The time series data collected in past and imported into WEKA has some rough data change trends and patterns. Recognize theses trends and patterns with the help of WEKA machine learning models can help to predict the future data. If we can predict the number of future remote user connections, it may help us to rebalance the working load in each machine timely and guide users the best host machine to use.

WEKA time series analysis and forecasting use "ahead step" as a time unit to predict the future data. This time unit is automatically detected by the WEKA model training process. When predicting future data with the generated forecast model, each time unit is considered as one "ahead step" so the future data prediction means predicting future data in one or more "ahead steps" based on past data change seasonality and change trends.

Two years of data (03/01/2017 through 03/01/2019) is used to train the forecast model. The time series analysis with different base learners are applied for forecasts of 1,2, and 6 time units (steps). The detected length of the time unit (i.e. ahead step) in this case is 2 hours 31 minutes and 31 seconds. Figure 2 shows the data using a one-step ahead data prediction model. Figure 3 shows the data with 1, 2, and 6 steps ahead data prediction.

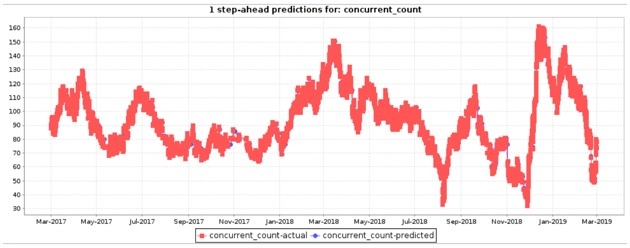


Figure 2: NX server connection count: data and data with one-step ahead prediction.

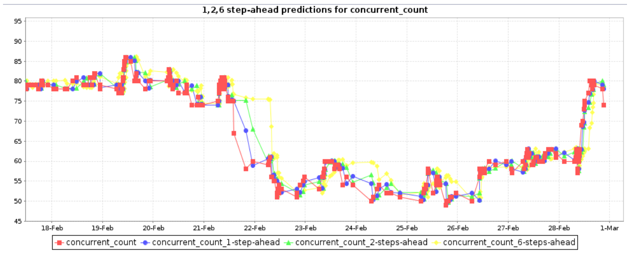


Figure 3: NX Server connection count data and data with 1, 2, and 6 steps ahead prediction.

Figure 2 and Figure 3 show that the predicted data are very close to the real data. As the number of ahead steps increases, prediction accuracy decreases. The percent of correctly predicted ranges of connection counts is used to measure accuracy. The Mean Absolute Error (MAE),

MAE = sum(abs(predicted - actual)) / N,

(where N is the number of data point evaluated), and Root Mean Squared Error:(RMSE),

RMSE=sqrt(sum((predicted - actual)^2) / N),

are also calculated for evaluation of prediction accuracy. For one step ahead prediction, the forecast model with linear regression gave 99% prediction accuracy.

The MAE data measured also indicates that the prediction accuracy drops as prediction ahead steps increases as shown in Fig. 4. The actual future data prediction with the forecast model also shows this trend.

The model makes pretty good prediction in the short term (1 or 2 ahead steps). As the future time (or ahead steps) increase, the prediction accuracy drops. This is mainly because the prediction error accumulates further into the future.

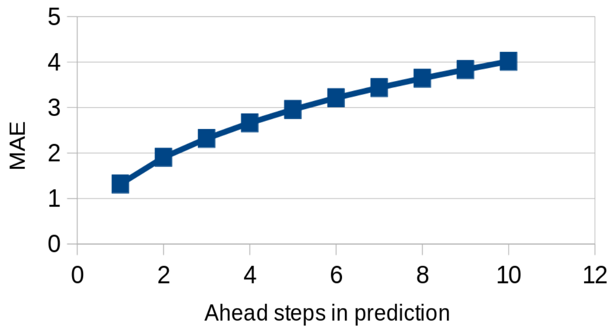


Figure 4: MAE vs ahead steps in prediction.

Table 1: MAE/RMSE Results of Different Base Learners (with one step ahead prediction)

Base Learner	MAE	RMSE
Linear Regression	1.3168	1.945
Neural Network	2.1867	2.603
SMO Regression	18.97	23.7
Gaussian Process	18.99	23.7

As the data shows in Table 1, the Linear regression base learner gives better accuracy. The Neural Network (Multilayer Perception) also gives good prediction accuracy. SMOreg (Sequential Minimal Optimization regression) and Gaussian process don't work well for this data. It is good practice to test many different base learning models with the same training data set and then select the best base model for the target data series forecast.

TIME SERIES FORECASTING IN APPS

With the Java WEKA API, the time series analysis and forecasting model can be implemented in controls applications. The forecasting model can then be used for live IO data prediction with one or more pre-configured ahead steps and selected base learner. This can be useful for accelerator control system for taking proactive feedback/feedforward actions based on forecasting values.

In the RHIC controls system, the humidity and temperature of RHIC ring service buildings are always monitored and logged to ensure accelerator controls devices in the service buildings are in good working environments. Forecasting the humidity and temperature changes ahead of time can help the accelerator controls system operators take precautionary measures before a climate related failure occurs.

We create a generic time series data forecasting program by implementing WEKA's Time Series Forecast API in a Java program named TSF. This program can analyse past logged data or monitor the live data along with prediction data and display the original data along with the predictions in a live display. The program is flexible in specifying how many ahead steps is used to predict the future data based on the historical data and which base learner to use. The TSF program supports the following features:

- Select any target parameter of any device in RHIC controls system as the target attribute set to be analysed and predicted.
- Select any associated parameter or device as the associated attribute set;
- Monitor live data from selected devices, and analyse and predict the target data set from the streamed IO.
- Loading previously logged data;
- Change the ahead step number (i.e the time ahead) for data prediction;
- Select different base machine learners;
- Display program detected periodicity of data changes in the source data;
- Re-analyse data with different program settings (such as after the ahead step number or base learner is changed);

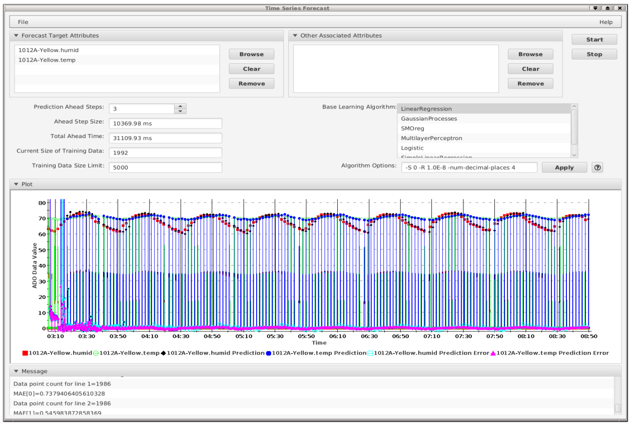
The forecasting analysis algorithm in this program can be easily implemented in other accelerator controls programs for conducting all kinds of data mining tasks available in the WEKA system.

Figure 5 below shows the forecast analysis with logged temperature and humidity data in RHIC service building 1012 for the yellow ring. The IO data, along with the predicted data, and the prediction error at each data point are plotted. The prediction ahead steps was set to 3, representing about 31 seconds. The results show that the predicted values are pretty close to the actual values. The prediction accuracy increases more as the number of data samples (data training size) increases.

We also examined the effect of ahead steps on the prediction accuracy with MAE data, shown in Fig. 6. Each ahead step is about 10.37 seconds. Ahead steps ranging from 1 (~10 seconds) to 5 (~52 seconds) shows that the prediction accuracy is reasonably good. The

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA043

prediction errors increase significantly when the ahead steps is set to 6 or more.



1012A-Yellow.humid
1012A-Yellow.temp

Prediction Ahead Steps: 3
Ahead Step Size: 10369.98 ms
Total Ahead Time: 31109.93 ms
Current Size of Training Data: 1992
Training Data Size Limit: 5000
Base Learning Algorithm: LinearRegression
Algorithm Options: -S 0 -R 1.0E-8 -num-decimal-places 4

```
Data point count for line 1=1986
MAE[0]=0.7379406405610328
Data point count for line 2=1986
MAE[1]=0.545983872858369
```

Figure 5: Forecasting analysis of logged humidity and temperature data.

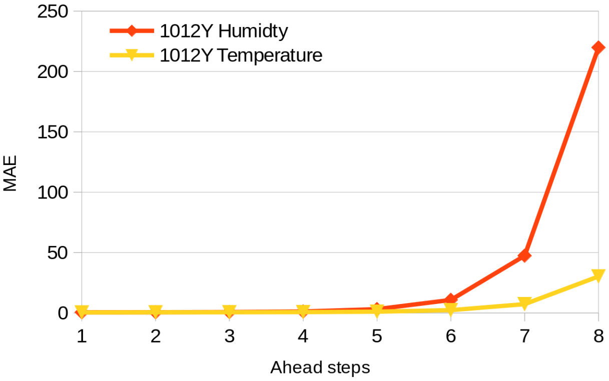
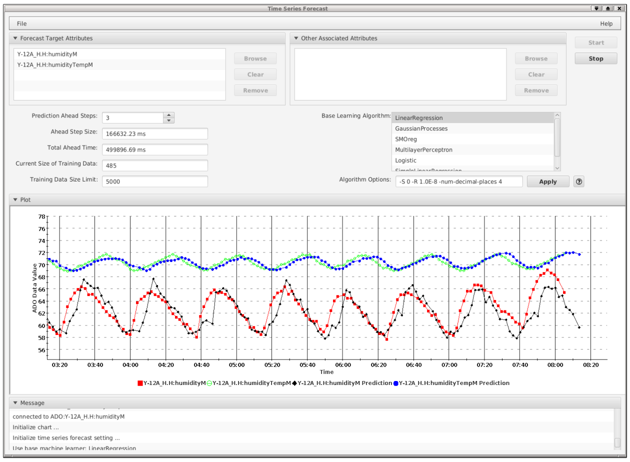


Figure 6: Prediction accuracy vs ahead steps.

Figure 7 below shows live IO data monitoring along with live data prediction using 3 ahead steps. The last three data points (black and blue lines) are the live predicted values. In this case, the total prediction ahead time is ~8 minutes (using three ahead steps). The time series forecast model does a reasonably good prediction in this case.

With this live prediction capacity, controls programs can take proactive actions such as sending an alarm when prediction values become abnormally high (or low). This can provide an operator more time to handle any necessary changes to climate controls.

The TSF application can be updated to become configurable for different types of IO data sources and connect to appropriate feedback/feedforward actions. These techniques can be applied to other controls applications leading to a more "intelligent" controls system.



Y-12A_H.H:humidityM
Y-12A_H.H:humidityTempM

Prediction Ahead Steps: 3
Ahead Step Size: 166632.23 ms
Total Ahead Time: 499896.69 ms
Current Size of Training Data: 485
Training Data Size Limit: 5000
Base Learning Algorithm: LinearRegression
Algorithm Options: -S 0 -R 1.0E-8 -num-decimal-places 4

```
connected to ADO:Y-12A_H.H:humidityM
Initialize chart ...
Initialize time series forecast setting ...
Use base machine learner: LinearRegression
```

Figure 7: Monitoring and forecasting of live IO data.

SUMMARY

Many accelerator controls system data are time series based, and many of them have detectable change trends or patterns. With the help of the Java based data mining package, WEKA, we are able to use WEKA's GUI tool and programming API to dynamically monitor time series data change trends and patterns, and predict future data changes accordingly. This gives related accelerator controls systems a way to update certain controls parameters intelligently and proactively. The Data Management Tool (dmt) java application was updated to stream data in accelerator controls databases into the WEKA data mining engine. dmt handles the data conversion and takes advantage of WEKA's powerful data mining features to analyse any desired controls data for a target problem. WEKA's Java API was implemented for time series data analysis and forecast. The subsequently developed java program was used to dynamically analyse past logged data, as well as monitor live accelerator controls data from the RHIC controls system. The application successfully performed live data prediction by configuring a base-learner in advance of the analysis. This makes it possible to apply proactive actions to related controls parameters and realize a more intelligent controls system.

REFERENCES

[1] Weka 3: Machine Learning Software in Java, `https://www.cs.waikato.ac.nz/~ml/weka/`

[2] Attribute-Relation File Format (ARFF), `https://www.cs.waikato.ac.nz/~ml/weka/arff.html`

[3] Time Series Analysis and Forecasting with Weka, `https://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka`

[4] Time Series, `https://en.wikipedia.org/wiki/Time_series`