International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

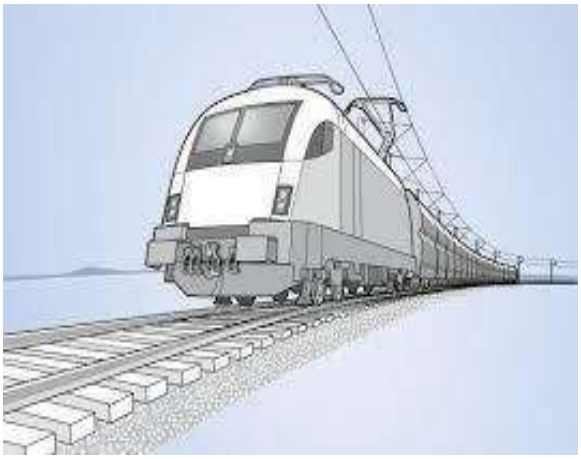
ONLINE FAILURE PREDICTION FOR RAILWAY TRANSPORTATION SYSTEM BASED ON FUZZY RULES AND DATA ANALYTICS

VINOTH KUMAR B[1], UDHAYA KUMAR G[1], SRIRAM M[1], Mr R VIDHYA PRAKASH[2]

[1]STUDENT, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, RMK ENGINEERING COLLEGE
[2]PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, RMK ENGINEERING COLLEGE

Abstract **–** Online failure prediction provides a way to actively manage faults. It can identify during runtime whether a failure would occur in the near future based on the assessment of the current state of the system. The available technologies for online failure prediction need some pre-cautional knowledge, such as the model of the system or failure patterns. In our system a new method based on fuzzy rules and time series analysis. , fuzzy rules are used to model the characteristics among different variables, whereas univariate time series analysis is used to describe the evolution of each variable. There are two predicted values for a dependent variable -one is from the ARIMA model, and the other is computed from fuzzy logic. A failure in some time period occurs when there is an exceeding difference between the two values beyond a threshold. The proposed system considers both the evolutionary trend of each variable and also the correlation among different variables. Moreover, we do not need any prior knowledge such as system model or failure patterns. As an example to illustrate our method we propose the railway transportation system. Online failure prediction is a way of short-term prediction made at runtime on the basis of information obtained by monitoring the running system.

Keywords—Autoregressive integrated moving average (ARIMA), failure prediction, fuzzy rules, railway transportation system, and time series analysis.



I. INTODUCTION

SAFETY-CRITICAL systems, such as railway control systems, may use off-the-shelf hardware and software developed by different companies. This increases the probability of the occurrence of faults, which may be propagated to others, raising the risk to stagnate the entire system. With the increasing growth of the complexity and the dynamicity of computer systems, classical reliability cannot reflect the real status of the systems. Online failure prediction provides a way to actively manage faults. It can identify during runtime whether a failure would occur in the near future based on the assessment of the current state of the system.

The goal of online failure prediction is to forecast, while the system is running, whether a failure may occur at some time in the future. Specifically, at present time t, the potential occurrence of a failure is predicted with the lead time Δtl based on the data of the system monitored within a data window of length intends to predict reliability metrics, such as failure rates, from the characteristics of software or development process. The main difference between them is that software reliability prediction

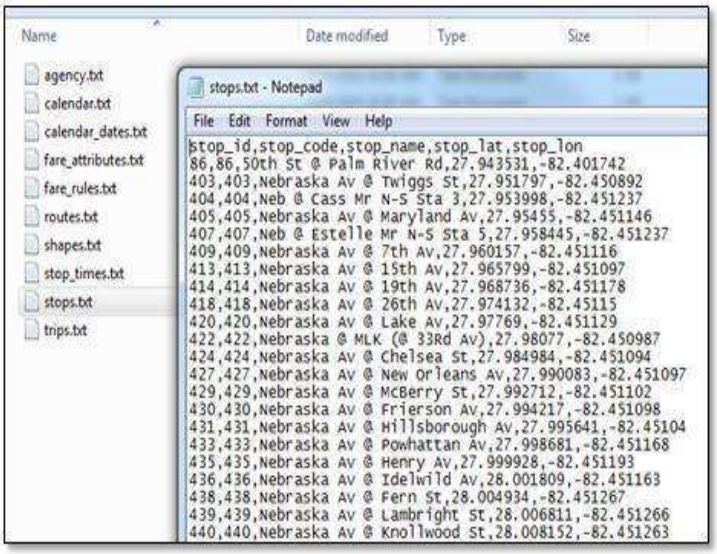
deals with long-term forecast, whereas online failure prediction deals with comparatively short time intervals and must be conducted at the current system state.

II. OVERVIEW

This overview consists of four sections

- Dataset fetching
- Preprocessing
- Estimation
- Prediction

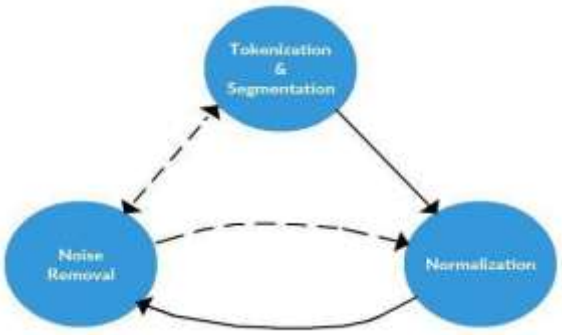
2.1 DATASET FETCHING



Three datasets are involved for estimation and prediction. The GPS dataset contains the GPS coordinates of every train every 20-40 seconds. The smart card dataset records every smartcard's users' boarding event. The train route map is static data and used for mapping the passenger's boarding event to the station. We collect these three datasets to localize the boarding events.

2.2 PREPROCESSING

In preprocessing phase the symbolic attributes are converted to numerical attribute. Also we filter the noise data by filtering missing attribute. Data are converted to format need for estimation phase. In this phase symbols are converted into numerical and filtered.



2.3 ESTIMATION PHASE

We further propose a probabilistic model to estimate the number of the alighting passengers at each passed station. Therefore, with both numbers of the boarding and alighting passengers, we finally obtain an estimated value of the number of the on board passengers.

2.3.1 Time Synchronization and Boarding Event Localization

To label a smart card payment record with a station, we match the time stamps in AFC records and OBU records .We then can locate the stations of AFC records by matching them with the OBU records.

2.3.2 Estimation of L (I , j)

Based on above time synchronization and boarding event localization process, we can then label every AFC payment with its corresponding station.

2.3.3 Estimation of N (I , j)

With the number of getting on passengers known, the next task is to estimate the number of passengers getting off at each station. Then we are able to output the number of passengers on the bus

2.4 PREDICTION PHASE

To predict the passenger flow, we firstly search the historical data to find the passenger flow pattern that is most similar to current estimation.

2.4.1 Calibration Based on Extended Kalman Filter

We use predictor to calibrate the coarse prediction and output the final predictive value

III LITERATURE SURVEY

There are many technologies that have been proposed, and the typical ones are summarized as follows.

3.1 MARKOV MODEL BASED METHOD

Vaidyanathan and Trivedi construct a semi-Markov reward model in order to obtain a workload-based estimation of the resource consumption rate. The model is then used to predict the time for resource exhaustion. . Daidone et al use a hidden Markov model (HMM) to infer whether the true state of a monitored component is healthy or not. Some extensions of the HMM are also proposed. For example, Zheng et al propose a failure prediction approach based on cloud theory and HMM. This approach expands the HMM by training the model with cloud theory. Baldoni et al propose an architecture with HMM to predict failures in an air traffic control system. Salfner and Malek use the hidden semi-Markov model for online failure prediction.

3.2 BAYESIAN MODEL BASED METHOD

Csenki applies the Bayesian prediction analysis technique to the Jelinski–Moranda software reliability model. This method can yield an improved estimation based on the failure probability distribution function since it benefits from the knowledge obtained from previous failure occurrences in a Bayesian framework. Bai et al use a Markov Bayesian network model for software failure prediction

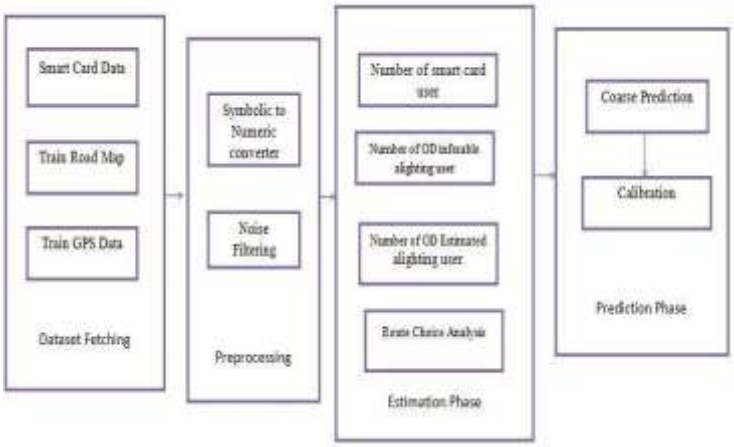
3.3 NEURAL NETWORK BASED METHOD

Troudet et al use neural network to predict failures of the mechanical parts. Neville describes how the standard neural network can be used for failure prediction in large-scale engineering plants. Fu and Xu build a neural network to approximate the number of failures in a given time interval.

3.4 RULE BASED METHOD

Berenji et al use Gaussian rules, a so-called diagnostic model, to compute a diagnostic signal, whose value ranges from 0 (fault free) to 1 (faulty), based on the input and output data of components. The rule base is algorithmically derived by means of clustering training data, which consists of input/output pairs in both faulty and fault-free cases. Kiciman and Fox construct a decision tree from runtime paths in order to identify faulty components. Kolarik et al propose a fuzzy-rule-based method to predict the human performance reliability. By using fuzzy rules to model the relationship between failure modes and performance metrics, this approach can do online performance monitoring, real-time performance forecasting, and reliability assessment.

IV. System Architecture



V. WORKING

The working methodology is explained in two different categories

- Time Series Analysis
- Fuzzy Logic

5.1 TIME SERIES ANALYSIS

During the running of the software system, the collected data for a single variable are a time series. There are several methods such as exponential smoothing model and ARIMA model, that are available for prediction of time series1) generalizing random walk models fine-tuning to eliminate all residual autocorrelation;2) generalizing exponential smoothing models incorporating long-term trends and seasonality; and3) stationarizing regression models using lags of the dependent variables and/or lags of the forecast errors as regressors. It easy implementation but great prediction power, the ARIMA model is selected to predict univariate time series that can be stationarized by transformations.

Step 1: Checking stationarity of the time series. To check the stationarity of a series, we often use augmented Dickey–Fuller (ADF) unit root testing. If there exists a unit root, the series is non stationary. Note that to do the ADF testing ,we should first determine the order p of the AR model. There are no deterministic ways to select such a value of p. Once a stationary series is obtained, stop the procedure. The total number of differences is the required value of d.

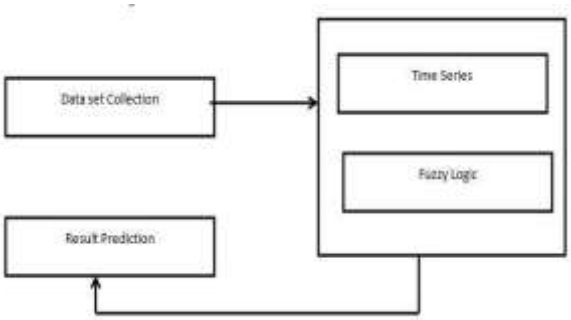
Step 2: Determining model parameters. We need to set the values of parameters p, q, and the coefficients after the stationarity process for the time series. We first determine the upper bounds of p and q. This can be done with guidance of the plots of ACF and partial autocorrelation function (PACF) . For each pair (p, q), we can compute the AR and MA coefficients with the stationary time series. Then, we select a proper pair (p, q). This process is actually a process to select an asymptotically

optimal model. There are many criteria that can be used to select an asymptotically optimal model, among which two most popular and widely used are Akaike information criterion (AIC) and Bayesian information criterion (BIC) . Usually, AIC focuses more on the asymptotic efficiency, whereas BIC on the asymptotic consistence. Thus, for AIC, we want to find a model that maximizes predictive accuracy, whereas for BIC, we expect to find the correct model.

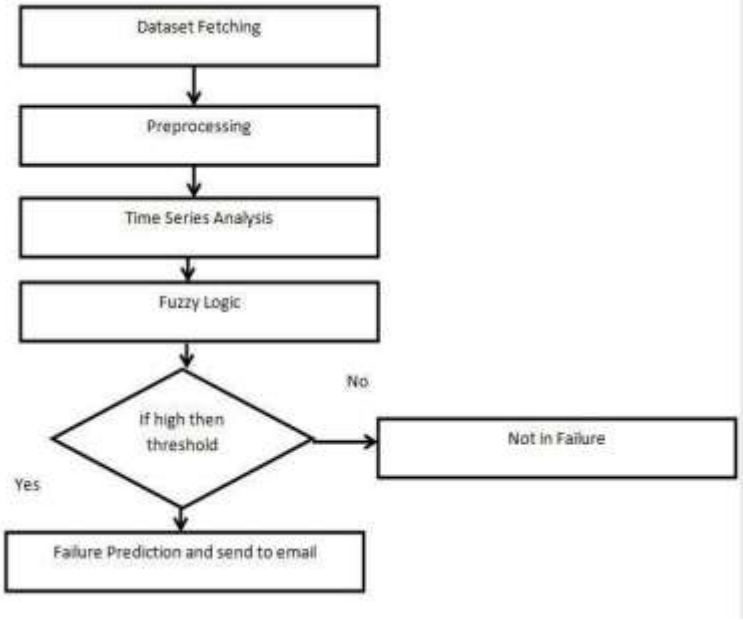
Step 3: Forecasting. Once we have the optimal prediction model ARIMA(*p, d, q*), we can forecast the monitored variable's .In this algorithm, function *Adjust* is applied to stationarize the series and returns to the number of differences, *Model Identify* determines the coefficients of the ARIMA model for each pair (*p, q*), and *Parameter Estimation* returns the model *ARIMA*(*p, q*) with minimal AIC value. There are two kinds of prediction functions in this algorithm. The first one, *Ipm*(*Zt*), in function *Error Analysis* is used to test the obtained model. It predicts the values of test data, and then the function *Error Analysis* checks whether the sequence of the prediction error is white noise. The second one, *Epm*(*Zt*),is used to predict the future value that we are interested in.

5.2 FUZZY LOGIC

We generate fuzzy rules based on the algorithms. To better serve the algorithm, we need to first normalize the raw data. The most commonly used normalization methods are: min–max standardization, Z-score standardization, and standardization decimal scaling. A single fuzzy if-then rule (or simply fuzzy rule) has the form Ri: IF x is A THEN y is B where x and y are two linguistic variables whose domains of discourse are X and Y , respectively; A and B are their possible terms, respectively.



VI. FLOW DIAGRAM



VII. CONCEPTS

For this the concepts used in the application development depends on the task we implement:

- JAVA
- JAVA API
- SOURCE CODE
- NETBEANS 8.1
- SMTP
- ODBC
- SQLYOG
- ARIMA –TIME SERIES ANALYSIS
- FUZZY LOGIC

VIII. CONCLUSION

In this paper, we present a method combining fuzzy rules and ARIMA model for online failure prediction. The main idea is that given a set of future time instants, we predict the values of a monitored variable from fuzzy inference and the ARIMA model, respectively. Then for each time instant, we compare these two values: If the difference is larger than a given threshold, we conclude that a failure would occur at that time; otherwise, the system can run well at that time. By combining the time series analysis and fuzzy rules, our method can avoid spurious regression, which is a key problem in statistic analysis

IX. REFERENCES

1) [1]Zuohua Ding ,yuan Zhou and Geguang Pu—Online Failure Predictin For Railway Transportation Systems Based on Fuzzy Rules and Data Analysis‖.

2) K. Aho, D. Derryberry, and T. Peterson, —Model selection for ecologists: The worldviews of AIC and BIC,‖ Ecology, vol. 95, no. 3, pp. 631–636, Mar. 2014.

3) H. Akaike, —Information theory and an extension of the maximum likelihood principle,‖ in Proc. 2nd Int. Symp. Inf. Theory, Budapest, Hungary, 1973, pp. 267–281.

4) Amin, A. Colman, and L. Grunske, —An approach to forecasting QoS attributes of web services based on ARIMA and GARCH models,‖ in Proc. IEEE 19th Int. Conf. Web Serv., Honolulu, HI, USA, Jun. 2012, pp. 74–81.

5) C. G. Bai, Q. P. Hu, M. Xie, and S. H. Ng, —Software failure prediction based on a Markov Bayesian network model,‖ J. Syst. Softw., vol. 74, vol. 3, pp. 275–282, Feb. 2005.

6) R. Baldoni, L. Montanari, and M. Rizzuto, —On-line failure prediction in safety-critical systems,‖ Future Gener. Comput. Syst., vol. 45, pp. 123– 132, Apr. 2015.

7) H. Berenji, J. Ametha, and D. Vengerov,—Inductive learning for fault diagnosis,‖ in Proc. IEEE 12th Int. Conf. Fuzzy Syst., St. Louis, MO, USA, May 2003, pp. 726–731.

8) K. Y. Cai, L. Cai, and W. D. Wang, —On the neural network approach in software reliability modeling,‖ J. Syst. Softw., vol. 58, no. 1, pp. 47–62, Aug. 2001.

9) J. L. Castro, —Fuzzy logic controllers are universal approximators,‖ IEEE Trans. Syst., Man, Cybern., vol. 25, no. 4, pp. 629–635, Apr. 1995.

10) A. Csenki, —Bayes predictive analysis of a fundamental software reliability model,‖ IEEE Trans. Rel., vol. 39, no. 2, pp. 177–183, Jun. 1990.