The Comparison of Holt – Winters and Box – Jenkins Methods for Software Failures Prediction

Vitaliy Yakovyna and Oleksandr Bachkai

Lviv Polytechnic National University, Lviv 79013, Ukraine
vitaliy.s.yakovyna@lpnu.ua, abachkay@gmail.com

Abstract. Software reliability has proven to be a very important issue in several past decades. Software failures may result in a low-quality product, which is probably unacceptable for customer and other stakeholders. So that, additional works and budget will be required to rise the reliability to an appropriate level. This could be avoided by managing software quality throughout the whole development process. In order to manage the resources in optimal way, project coordinator might need some quantitative indicators, like predicted number of failures, which can be obtained by forecasting. In case of software failures, the most common forecasting approaches include software reliability models. Their main disadvantage is that they do not fit all different types of software, because they rely on the assumptions about software properties and behavior. This paper deals with statistical time-series forecasting techniques, which depend only on the previous values, and hence can be the ultimate solution for failure prediction of different types of software. The Holt – Winters smoothing model and ARIMA regression model were considered to predict the Angular software failures on a weekly basis. It is shown that while the performance of ARIMA model is almost twice worse than that of Holt – Winters model, the accuracy is almost an order of magnitude better.

Keywords: Software Reliability, Time-Series, Failures Prediction, Holt – Winters Model, ARIMA model.

1 Introduction

Today the process of software development is highly imposed by certain limitations on cost and time, as well as requirements for quality and reliability. Many organizations, involved in software development, spend large amount of their funds on testing and refactoring in order to prevent failures.

The greatest problem facing the industry today is how to assess quantitatively software reliability characteristics (see e.g. [1, 2] and others). Research on software reliability engineering has been conducted during the past three decades and numerous statistical models have been proposed for estimating software reliability. Most existing models for predicting software reliability are based purely on the observation of software product failures where they require a considerable amount of failure data to obtain an accurate reliability prediction. Some other research efforts have recently developed reliability models addressing fault coverage, testing coverage, and imperfect debugging processes [1].

The later a bug is discovered, the more expensive and difficult it is to resolve. This is extremely valuable in case of Software as a Service or cloud software development, which show constantly growing share of the market during the last decade. Quality control methods such as inspection and testing aim to detect faults prior to release. Unfortunately, code inspection and testing are costly in terms of time and manpower, so managers seek to optimize their effectiveness. Bug prediction has been suggested as a means to this end (see e.g. [3–5]). The growth of empirical software engineering techniques has led to increased interest in bug prediction algorithms [3].

2 Related Works Analysis

Forecasting methods are commonly divided into two main groups: intuitive and formalized (Fig. 1) [6].

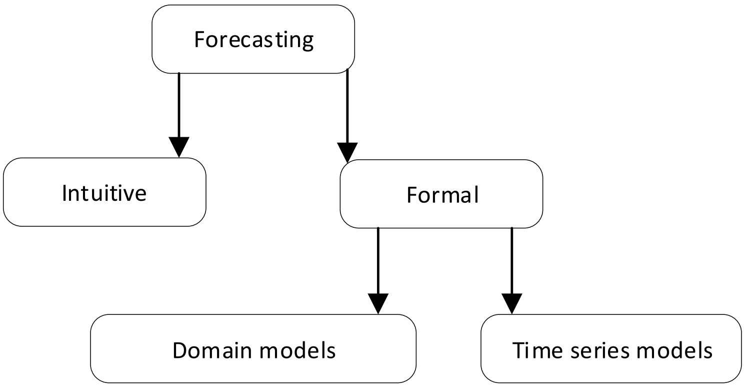


Fig. 1. Classification of forecasting methods.

Intuitive forecasting methods include expert judgments and estimates. Today, they are often used in marketing, economics, politics and other domains, which behavior are very complex or difficult to predict with mathematical models [7].

Formalized methods are the methods, which use mathematical models to predict future values. They are divided into domain models and time-series models.

Domain models – models, based on processes, rules and mechanisms of domain. For example, weather forecast model contains an equation of dynamics of fluids and thermodynamics. In the context of software failures prediction the most common approaches include software reliability models. Their main disadvantage: they do not fit all different classes of software, because they depend on particular aspects of it [1]. In order to create an adequate model of software reliability and be able to make decisions, based on such a model, a deep understanding of the processes, methodologies and technologies of creation and testing of software is required.

Time-series models are mathematical models of forecasting, which seek to find the dependence of the future value from the past value within the process itself and calculate the prediction, based on this dependence. These models are universal for various domains, that is, their general appearance does not change, depending on the nature of the time series [8]. Time series models [9] can be further divided into (see Fig. 2):

- regression models;
- smoothing models;
- models based on neural networks.

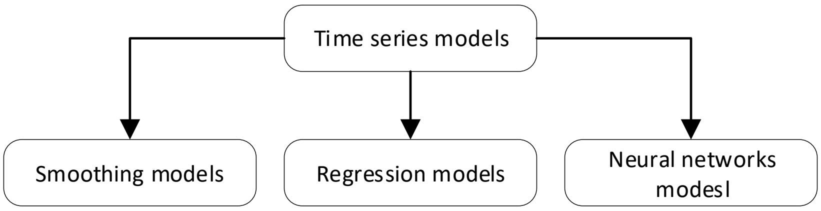


Fig. 2. Time series models.

Time-series prediction is based on different models and approaches and is widely used for modelling various aspects of human activity [10–12]. The Holt – Winters forecasting is one of the most used among the smoothing models. This forecasting procedure is a variant of exponential smoothing which is simple, yet generally works well in practice, and is practically suitable for producing short-term forecasts time-series data (see e.g. [13, 14]). In [15] it was shown that Holt – Winters short-term model is equivalent to particular ARIMA model, and generally do not lie within that subset of the ARIMA class which forms the basis of the Box – Jenkins modelling approach. It is argued that the models considered in [15] have a reasoned structure, and are to be preferred to the Box – Jenkins models for most socio-economic applications.

On the other hand, Auto-Regressive (AR) models were first introduced by in 1926 and subsequently supplemented in 1937 by Moving Average (MA) schemes [16]. Wold [17] combined both AR and MA schemes and showed that ARMA processes can be used to model a large class of stationary time series as long as the appropriate order of p, the number of AR terms, and q, the number of MA terms, was appropriately specified. The paper [16] concludes that the major problem of ARIMA models is the way of making the series stationary in its mean that has been proposed by Box and Jenkins. In addition, it was shown that using ARMA models to seasonally adjusted data slightly improves post-sample accuracies while simplifying the use of ARMA models [16].

These two classes of forecasting techniques have been compared in number of empirical studies (see e.g. [18–20]). Thus, in [18] the forecasting approach for short- and long-term heat load forecasting on the three levels: monthly, weekly and daily forecasting bases was presented. Based on chosen accuracy measures, Multiple regression was recognized as the best forecasting method for daily and weekly short-term heat load forecasting, whereas Holt–Winters methods ensured the best forecasting values in purpose of long-term heat load forecasting and monthly short-term heat load forecasting

[18]. Paper [19] determines the forecasting accuracy of Holt – Winters and ARIMA models for samples of telemarketing data, and concludes that ARIMA models with intervention analysis perform better for the time series studied. Paper [20] uses intraday electricity demand data from ten European countries as the basis of an empirical comparison of univariate methods for prediction up to a day-ahead. The ARIMA and principal component analysis methods performed well, but the method that consistently performed the best was the double seasonal Holt – Winters exponential smoothing method [20].

Promising methods of software failures prediction are methods based on nonparametric models [21, 22]. Such models do not have main drawbacks and difficulties of analytical models because they do not make any assumption about the mechanism of software failures. Besides smoothing and regression models, the artificial neural networks are widely used for software failures prediction because of their proven quality of generalization and approximation of almost any smooth functions [23]. In [24] the study of the efficiency of software failures time-series prediction by RBF neural networks was presented. The achieved root-mean-square error (RMSE) was as low as 1.3% [24].

Thus, the performed related works analysis shows that for different domains with the peculiar time-series features different forecasting methods could perform better. Hence, the goal of this paper is to compare Holt – Winters and ARIMA forecasting for software failures time-series.

3 The Models Description

A time series is a series of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time [6, 10].

There are many different models of time series forecasting, but all they are aimed to investigate the following three components [25], shown in Fig. 3:

- Seasonal: patterns that repeat with a fixed period of time.
- Trend: the underlying trend of the metrics.
- Random: also call “noise”, “irregular” or “remainder”, this is the residuals of the original time series after the seasonal and trend series are removed.

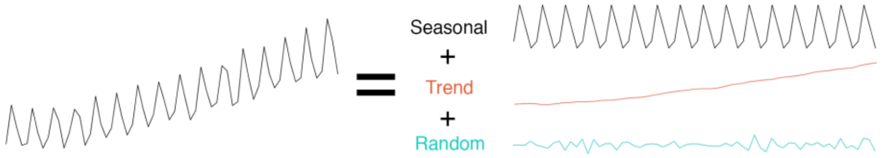


Fig. 3. Trend, seasonal and random components.

3.1 Holt – Winters Model

Smoothing methods are used to reduce the effect of random oscillations in time series. They give the opportunity to receive “pure” values that consist only of deterministic components [9].

The most advanced method of this group is Holt – Winters method, which is also called triple exponential smoothing [13–15]. Let the observed time series be denoted by $y_1, y_2, \dots, y_n$. A forecast of y_{t+h} based on all of the data up to time t is denoted by $\hat{y}_{t+h|t}$. The model then is described by the forecast equation (1), which includes the level (2), trend (3), and seasonal (4) components.

$$\hat{y}_{t+h|t} = l_t + b_t h + s_{t-m+h_m^+}, \tag{1}$$

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}), \tag{2}$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}. \tag{3}$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}. \tag{4}$$

Here m is the length of seasonality, l_t represents the level of the series, b_t denotes the growth, s_t is the seasonal component, $\hat{y}_{t+h|t}$ is the forecast for h periods ahead, and $h_m^+ = [(h-1) \bmod m] + 1$.

3.2 ARIMA model

The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. Lags of the stationarized series in the forecasting equation are called “autoregressive” terms, lags of the forecast errors are called “moving average” terms, and a time series, which needs to be differenced to be made stationary, is said to be an “integrated” version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models [8, 16].

A non-seasonal ARIMA model is classified as an $ARIMA(p, d, q)$ model, where:

- p is the number of autoregressive terms,
- d is the number of non-seasonal differences needed for stationarity, and
- q is the number of lagged forecast errors in the prediction equation.

To fit parameters a methodology was invented by Box and Jenkins [8]. It consists of three steps.

1. Model Identification. Use plots and summary statistics to identify trends, seasonality, and auto-regression elements to get an idea of the amount of differencing and the size of the lag that will be required.
2. Parameter Estimation. Use a fitting procedure to find the coefficients of the regression model.
3. Model Checking. Use plots and statistical tests of the residual errors to determine the amount and type of temporal structure not captured by the model.

The main approaches to fitting Box–Jenkins models are nonlinear least squares and maximum likelihood estimation. Maximum likelihood estimation is generally the preferred technique.

There is also a seasonal version of ARIMA It incorporates both non-seasonal and seasonal factors in a multiplicative model. One shorthand notation for the model is $ARIMA(p, d, q) \times (P, D, Q)S$, with p – non-seasonal AR order, d – non-seasonal differencing, q – non-seasonal MA order, P – seasonal AR order, D – seasonal differencing, Q – seasonal MA order, and S – time span of repeating seasonal pattern [8, 16].

4 Experimental

To carry out the research a desktop software application was developed, using Visual Studio and C#. Non-seasonal Holt-Winters and ARIMA models were implemented as separate modules (for reusability purposes). Besides, the module to calculate accuracy and performance of prediction method was implemented as well.

To receive a realistic estimation of forecasting methods, the failures data from real software project were used. They input data were obtained from the GitHub bug-tracking system. This study deals with the Angular project failure dates: 8 494 failures from October 2014 to January 2016 were fetched from GitHub, then compiled into 120 time series as number of failures per week.

Then, the initial 90% of time-series were selected as previous values, while the last 10% were used to compare the predicted and actual software failures values.

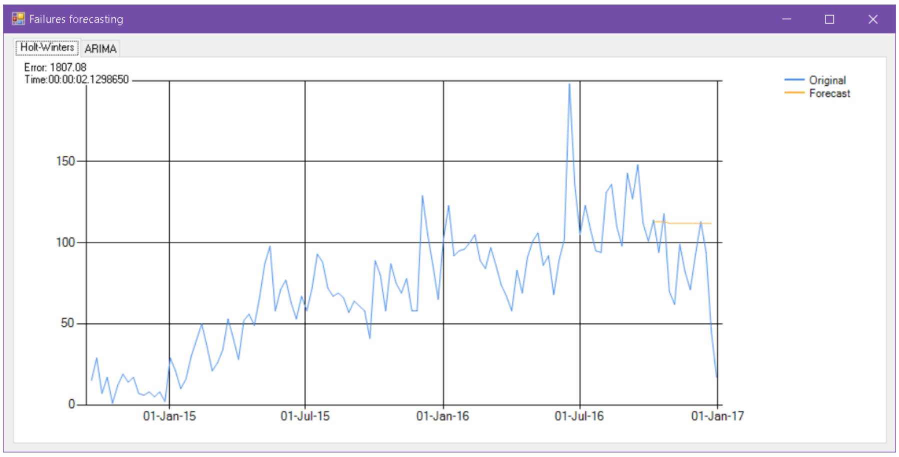
To estimate the prediction efficiency the RMSE was used as an accuracy measure, while execution time was used as a performance measure.

5 Results and Discussion

The studied Angular failures time-series is plotted in Fig. 4. As it can be seen from the Fig. 4, there is clear seasonal component in this time-series. Using smoothing and regression techniques as well as neural networks for such irregular data with seasonal component is not very effective and result in large approximation and prediction errors. However, there are several methods of increasing forecast accuracy. As mentioned in [9], if time series are compiled in a manner of cumulative sums (i.e. each time interval has sum of all previous values), a trend will be easier to estimate, and accuracy of forecast might be better. This is confirmed also by the conclusions of the paper [24]. Hence, the time-series were formed in a cumulative manner and models were used to predict software failures.

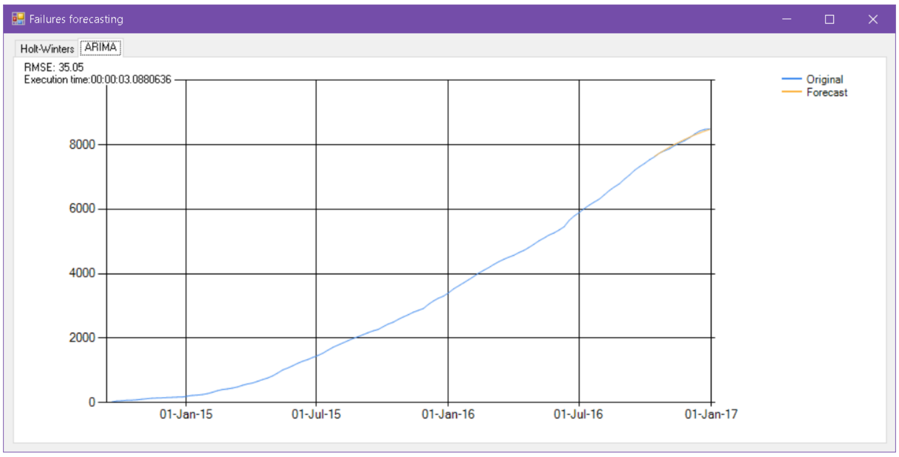
An example of the cumulative time series along with predicted using ARIMA model and actual data are shown in Fig. 5. As it can be easily seen, the time-series became smoother and the predicted software failures values are very close to the actual ones.

It should be noted, that non-seasonal versions of both models were implemented. Seasonal versions of models with non-cumulative time series should also give a better accuracy, since it is obvious from chart, that there is a seasonality in series. However, the efficiency of the seasonal versions of the models will be studied at the future work.



Error: 1807.08
Time:00:00:02.1298650

Fig. 4. Initial software failures time-series.



RMSE: 35.05
Execution time:00:00:03.0880636

Fig. 5. Results of forecasting of cumulative time-series using the ARIMA model.

Typical results of the prediction efficiency of Angular software failures using Holt – Winters and ARIMA models are listed in Table 1. While the performance of ARIMA model is almost twice worse than that of Holt – Winters model, the accuracy is almost an order of magnitude better. This could be explained by the range of the input data: the observed failures period is more than two years, while Holt – Winters model is best suited for short-range forecasting [18, 19].

Table 1. Results of software failures forecasting with different approaches.

Forecasting measure	Holt-Winters model	ARIMA model
Accuracy (RMSE)	220.64	35.05
Performance (execution time, seconds)	1.41	3.26

6 Conclusion and Future Work

Research on software reliability engineering has been conducted during the past three decades and numerous statistical models have been proposed for estimating software reliability. The growth of empirical software engineering techniques has led to increased interest in software failures prediction. The prediction of software failures is of large practical importance, because it provides project coordinators with some estimations, which they can use to manage and maintain software quality. Common ways of software failures prediction include usage of reliability models, which have a flaw – they cannot be fitted to all types of software. Time-series prediction methods, which are popular in economics domain, can solve this issue because they are do not rely on domain-specific processes and properties.

After analyzing different time series models two of them were selected as the most advanced ones: Holt – Winters smoothing model and ARIMA regression model. The software application was developed to study the efficiency of non-seasonal versions of these models for software failures forecasting. The case study was based on the Angular failures data obtained from the GitHub bug-tracking system, and included failures data for more two years period. The time-series were presented as the cumulative number of failures detected at all time intervals till the current one. To evaluate the efficiency of prediction two parameters were used: the RMSE as an accuracy measure, and the execution time as a performance measure.

The obtained results show that Holt – Winters model has better performance, while the ARIMA model was substantially better in sense of the prediction accuracy.

The future work will be devoted to studying the efficiency of the seasonal versions of the models, because the studied time-series have explicitly visible seasonality. Another study will be devoted to the influence of the forecasting interval on the efficiency of software failures prediction as well as studying different types of software both on long- and short-range intervals.

References

1. Pham, H.: System Software Reliability. Springer series in reliability engineering, Springer-Verlag London Limited (2006).
2. Trivedi, K. S., Bobbio, A., Muppala, J. K.: Greenbook: Reliability and Availability Engineering: Modeling, Analysis and Applications. Cambridge University Press (2017).
3. Rahman, F., Posnett, D., Hindle, A., Barr, E., Devanbu, P.: BugCache for Inspections: Hit or Miss? In: Proc. of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering, pp. 322–331, ACM, Szeged, Hungary (2011). DOI: 10.1145/2025113.2025157
4. Lewis, C., Lin, Z., Sadowski, C., Zhu, X., Ou, R., Whitehead Jr., E. J.: Does Bug Prediction Support Human Developers? Findings from a Google Case Study. In: Proc. of the Int. Conf. on Software Engineering ICSE'13, pp. 372–381, IEEE, San Francisco, CA, USA (2013). DOI: 10.1109/ICSE.2013.6606583
5. Sunghun Kim: Adaptive bug prediction by analyzing project history. PhD thesis, Univ. of California, Santa Cruz, (2006).

6. Abraham, B., Ledolter, J.: Statistical Methods for Forecasting. John Wiley & Sons, New York, NY, USA (2005).
7. Tikhonov, E. E.: Methods of forecasting in market environment: textbook. Nevinnomyssk (2006). (in Russian)
8. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M.: Time Series Analysis: Forecasting and Control. 5th edn. John Wiley & Sons, New York, NY, USA (2015).
9. Ajstrakhanov, D. D., Pugachova, M. V., Stepashko V. S. et al.: Conceptual basics of statistical monitoring. Derzhkomstat Publishing, Kyiv (2003). (in Ukrainian)
10. Weigend, A.: Time Series Prediction. Routledge, New York, NY, USA (2018).
11. Masters, T.: Neural, Novel and Hybrid Algorithms for Time Series Prediction. John Wiley & Sons, New York, NY, USA (1995).
12. Brown, R. G.: Smoothing, Forecasting and Prediction of Discrete Time Series. Dover Publications, Mineola, NY, USA (2004).
13. Chatfield, C.: The Holt – Winters forecasting procedure. Appl. Statist. 27(3), 264–279 (1978). DOI: 10.2307/2347162
14. Chatfield, C., Yar, M.: Holt – Winters forecasting: some practical issues. The Statistician 37, 129–140 (1988). DOI: 10.2307/2348687
15. Roberts, S.A.: A General Class of Holt-Winters Type Forecasting Models. Management Science 28(7), 808–820 (1982). DOI: 10.1287/mnsc.28.7.808
16. Makridakis, S., Hibon, M.: ARMA Models and the Box–Jenkins Methodology. Journal of Forecasting 16(3), 147–163 (1997). DOI: 10.1002/(SICI)1099-131X(199705)16:3
17. Wold, H.: A Study in the Analysis of Stationary Time Series. Almgrist & Wiksell, Stockholm (1938).
18. Tratar, L. F., Strmčnik, E.: The comparison of Holt–Winters method and Multiple regression method: A case study. Energy 109, 266–276 (2016). DOI: 10.1016/j.energy.2016.04.115
19. Bianchi, L., Jarrett, J., Hanumara, R. C.: Improving forecasting for telemarketing centers by ARIMA modeling with intervention. International Journal of Forecasting 14(4), 497–504 (1998). DOI: 10.1016/S0169-2070(98)00037-5
20. Taylor, J. W., McSharry, P. E.: Short-Term Load Forecasting Methods: An Evaluation Based on European Data. IEEE Transactions on Power Systems 22(4), 2213–2219 (2007). DOI: 10.1109/TPWRS.2007.907583
21. Khoshgoftaar, T. M., Szabo, R. M.: Predicting software quality, during testing, using neural network models: A comparative study. International Journal of Reliability, Quality and Safety Engineering 1, 303–319 (1994). DOI: 10.1142/S0218539394000222
22. Zheng, J.: Predicting software reliability with neural network ensembles. Expert Systems with Applications 36, 2116–2122 (2009). DOI: 10.1016/j.eswa.2007.12.029
23. Paliwal, M., Kumar, U. A.: Neural networks and statistical techniques: A review of applications. Expert Systems with Applications 36, 2–17 (2009). DOI: 10.1016/j.eswa.2007.10.005
24. Yakovyna, V. S.: Software failures prediction using RBF neural network. Odes'kyi Politechnichnyi Universytet. Pratsi 2(46), 111–118 (2015).
25. Extracting Seasonality and Trend from Data, https://anomaly.io/seasonal-trend-decomposition-in-r, last accessed 2017/05/01.