An ARIMA-Based Windowing Model for Software Reliability

Priscila Silva[1], Mini Kusum Paudel[1], Vidhyashree Nagaraju[2], and Lance Fiondella[1]
[1]Electrical and Computer Engineering, University of Massachusetts Dartmouth, MA, USA
[1]{psilva4, mpaudel, lfiondella}@umaasd.edu, [2]vidhyashreenagaraju@gmail.com

***Abstract*—Software reliability growth models (SRGMs) are crucial to predict failure rates during software testing to enhance performance and mitigate risks before deployment. Many SRGMs rely on the non-homogeneous Poisson process (NHPP), which face difficulties when dealing with non-stationary data and short-term fluctuations. Consequently, researchers have been applying time series modeling techniques such as the Auto-Regressive Integrated Moving Average (ARIMA) model to capture sudden anomalies in software reliability over time. However, ARIMA relies on the assumption of linearity within the predicted time series data, which diverges from the evident non-linearity observed in many instances of software failures. To address this limitation, this paper integrates a windowing technique into ARIMA models (ARIMA-W) to mitigate the impact of non-linearity in software failure observations by allowing them to continuously update their parameters based on recent observations. The models are assessed with a dataset containing the number of software failures over 181 intervals, and goodness-of-fit measures are computed to compare the performance of ARIMA and ARIMA-W. Results indicate that ARIMA tracked the number of software failures well but failed to characterize new instances when data used for model development was limited. However, the ARIMA-W exhibited significant enhancements across all goodness-of-fit metrics, specifically in predictive root mean squared errors ($PRMSE$), when more data becomes available for model fitting. Notably, when 70% and 80% of the data are utilized for model fitting, ARIMA-W achieves approximately 5 and 8 times lower $PRMSE$, respectively, compared to ARIMA, indicating higher accuracy in predicting future software failures when fitted to a larger historical dataset.**

***Index Terms*—Software reliability, ARIMA, windowing technique, software failure predictions, goodness-of-fit measures**

I. Introduction

Software reliability growth models (SRGM) [1] hold a significant position in the realm of software reliability engineering [2], as they facilitate the prediction of failure rates during testing phases to enhance software performance and mitigate risks before deployment. Numerous SRGMs have been proposed based on the Non-homogeneous Poisson Process (NHPP) [3], [4] to estimate changes in failure intensity over time. However, NHPP faces challenges when dealing with non-stationary data and short-term fluctuations. Consequently, researchers have proposed to address these limitations with time series models [5], which offer a flexible framework well-suited for capturing sudden changes or anomalies in software reliability. Nonetheless, time series models typically presume linearity in the data to be predicted, which is not always observed in many software failure data. Therefore, alternative methodologies to overcome this challenge would contribute to a more accurate and reliable application of time series models, ultimately enhancing the predictive capabilities of SRGMs.

Different parametric forms of NHPP SRGM have been proposed [6]–[8] to estimate the number of failures remaining in a software system. However, NHPP SRGM often requires data to be stationary over time and assumes a failure intensity function that may not capture the complex patterns and dynamics present in real-world software failure data. As an alternative, time series models [9] were applied to adapt to diverse software reliability scenarios and potentially provide more accurate predictions in environments encountering non-stationary time series data, trends, and seasonality. For example, Auto-Regressive Integrated Moving Average (ARIMA) models were applied [10]–[12] to predict the number of software failures expected in the subsequent interval based on failure counts observed in past intervals during the testing phase of software systems. Moreover, alternative methods were explored to enhance ARIMA predictions for software reliability. Such methods include optimization of parameter estimation using genetic algorithms [13], integration of ARIMA and Support Vector Machines [14] models to capture data characteristics in linear and nonlinear patterns, or inclusion of seasonal patterns [15]–[17] to improve long-term predictions. Although ARIMA models offer advantages over SRGM NHPP models to predict software failures over time, estimating ARIMA parameters and selecting the appropriate model order can be challenging for large and complex software failure datasets involving nonlinear patterns, adding complexity to the model development and implementation.

To address these limitations, this paper employs a windowing technique [18]–[20] to enhance the performance of ARIMA models to predict the failure count of software systems. The windowing technique, which is popular in the areas of finance and traffic management, breaks down the time series data into smaller windows and applies the ARIMA model to each window individually. The models are assessed with a dataset containing the number of software failures over 181 intervals. Goodness-of-fit measures including $RMSE$, $PRMSE$, and r^2_{adj} are computed to compare the performance of traditional ARIMA versus the ARIMA-based windowing (ARIMA-W) model. Results indicate that the ARIMA-W model exhibited significant enhancements across all goodness-of-fit metrics, especially in $PRMSE$, when more data becomes available for model fitting, achieving at least 5 times lower $PRMSE$ compared to ARIMA. Thus, the windowing technique allows the model to capture local patterns and

non-linearities within each segment, improving the overall prediction accuracy.

The remainder of this paper is organized as follows: Section II reviews ARIMA modeling. Section III presents ARIMA models incorporating windowing technique. Section IV describes goodness-of-fit measures for model validation. Section V illustrates the proposed approaches using an actual software failure dataset. Section VI offers conclusions and identifies opportunities for future research.

II. ARIMA Modeling

Auto-Regressive Integrated Moving Average (ARIMA) [9] models are a class of time series forecasting that uses past observations, also known as lags, and their own forecasting errors to predict future values of the time series. These models are widely used in various fields, including economics, finance, and engineering for their simplicity, interpretability, and effectiveness in capturing the underlying patterns and dynamics of time series data. To apply time series models, the data must be stationary, without exhibiting trends, also known as seasonality, and possess a constant mean and variance over time. In cases where the data is non-stationary, it can be converted to stationarity with differentiation, which may require multiple differentiation steps. ARIMA models are defined by the number of lagged observations indicating the order of the auto-regressive component (p), the degrees of differentiation (d) required to make the series stationary, and the number of lagged forecast errors q to specify the order of the moving average component, denoted as ARIMA(p, d, q).

In the context of software reliability, the ARIMA predicts the number of software failures in future time intervals, given information on the number of failures discovered in the previous intervals as

$$\hat{FC}(i) = \beta_0 + \sum_{k=1}^{p} \beta_k FC(i-k) + \sum_{k=1}^{q} \theta_k \varepsilon(i-k) \qquad (1)$$

where FC is the failure count in interval i, β_0 is the baseline number of failures, β_k is the coefficient describing the number of failures in intervals $(i-p) \leq (i-k) \leq (i-1)$, and θ_k is the coefficient associated with k times steps prior to the present time step $(i-q) \leq (i-k) \leq (i-1)$ of a sequential white noise process (ε), which are statistically independent and normally distributed with zero mean and finite variance.

To identify numerical estimates of parameters β_0, β_k and θ_k contained in Equation (1), least squares estimation [21] is applied to determine the values of the parameters that minimize the disagreement between the actual FC data and the predictions $\hat{FC}$ in time interval i, which is computed as

$$\min \sum_{i=1}^{n-\nu} (FC(i) - \hat{FC}(i))^2 \qquad (2)$$

where n is the total sample size available, and ν is the observations not used for model fitting.

III. ARIMA-Based Windowing Model

Auto-Regressive Integrated Moving Average models incorporating windowing technique (ARIMA-W) [18] divide the $(n-\nu)$ time series data available for model fitting into m smaller windows of fixed length n_w to enable the model to capture local patterns and non-linearities within each segment. By treating each window as a separate time series of non-overlapping data points, ARIMA models are applied independently to each window of data to predict the number of software failures at each interval as follows:

(S.1) Failure count is predicted in the first window as

$$\hat{FC}_{w_1}(i) = \beta_0 + \sum_{k=1}^{p_1} \beta_k FC(i-k) + \sum_{k=1}^{q_1} \theta_k \varepsilon(i-k) \qquad (3)$$

where $\hat{FC}_{w_1}$ is the failure count prediction at interval i contained in the first window w_1 of data, and p_1 and q_1 are the orders of the auto-regressive and moving average components of the ARIMA model applied to the first window w_1, respectively. Least squares estimation is then applied as

$$\min \sum_{i=1}^{n_{w_1}} (FC(i) - \hat{FC}_{w_1}(i))^2 \qquad (4)$$

to estimate the parameters β_0, β_k and θ_k in the first window w_1, containing n_{w_1} observations.

(S.2) Failure count is predicted for the remaining windows as

$$\hat{FC}_{w_j}(i) = \beta_0 + \sum_{k=1}^{p_j} \beta_k \hat{FC}_{w_{(j-1)}}(i-k) + \sum_{k=1}^{q_j} \theta_k \varepsilon(i-k) \qquad (5)$$

where $\hat{FC}_{w_j}$ is the prediction in interval i contained in the window w_j $\left(n_{w_{j-1}} < i \leq n_{w_j}\right)$, for a total of $2 \leq j \leq m$ windows, and p_j and q_j are the orders of the auto-regressive and moving average components of the ARIMA model applied to the j^{th} window, respectively. To predict the failure count in the window w_j, predictions $\hat{FC}_{w_{(j-1)}}$ of the previous window are used as inputs of Equation (5), which allows the model to capture patterns and relationships in the most recent observations to forecast the next window of data. Least squares estimation is applied for each window with

$$\min \left(\sum_{i=n_{w_{(j-1)}}+1}^{n_{w_j}} (FC(i) - \hat{FC}_{w_j}(i))^2 \right) \qquad (6)$$

to estimate their individually predictors β_0, β_k and θ_k.

(S.3) The failure count is predicted for future instances in the $(n-\nu) \leq i \leq n$ intervals, which contain the ν observations not used for model fitting, utilizing the model parameters and the failure count ($\hat{FC}_{w_m}$) predicted in the last window w_m.

IV. Model Validation

Goodness-of-fit measures [22] offer an objective quantitative method to compare alternative models to evaluate their performance on a specific dataset. In most real-world scenarios, no single model performs best on all metrics. Therefore, model selection often involves subjective judgment and

decision-making, with a preference for models that achieve lower errors. Common goodness-of-fit measures applied to validate model predictions include the root mean squared error, predictive root mean squared error, and adjusted coefficient of determination.

Root mean squared error is calculated by fitting a model with $n - \nu$ observations and then computing the root mean squared difference between the actual FC observations and predicted $\hat{FC}$

$$RMSE = \sqrt{\frac{1}{(n-\nu)-\rho}\sum_{i=1}^{n}(FC(i) - \hat{FC}(i))^2} \quad (7)$$

where $(n - \nu) - \rho$ denotes the degrees of freedom, which represent the quantity of independent information available for variation. This is calculated by subtracting the number of parameters ρ in the model from the sample size $n - \nu$ used for model fitting. Lower values of $RMSE$ are preferred.

Predictive root mean squared error involves fitting a model with the initial $n - \nu$ observations, and subsequently calculating the sum of squares of the prediction residuals for the remaining ν observations that were not utilized in model fitting

$$PRMSE = \sqrt{\frac{1}{\nu}\sum_{i=(n-\nu+1)}^{n}(FC(i) - \hat{FC}(i))^2} \quad (8)$$

where a lower PRMSE value indicates greater predictive accuracy.

Adjusted coefficient of determination [23] is the proportion of the variation in the number of failures FC that is explained by the model according to

$$r^2_{adj} = 1 - \left(1 - \frac{SSY - SSE}{SSY}\right)\left(\frac{(n-\nu)-1}{(n-\nu)-\rho-1}\right) \quad (9)$$

where

$$SSY = \sum_{i=1}^{n-\nu}\left(FC(i) - \overline{FC}\right)^2 \quad (10)$$

is the sum of squared errors associated with the naive predictor $\overline{FC}$ computed as the mean of the first $n-\nu$ observations used for model fitting, and

$$SSE = \sum_{i=1}^{n-\nu}(FC(i) - \hat{FC}(i))^2 \quad (11)$$

is the sum of squared errors between actual $(FC(i))$ and predicted $(\hat{FC}(i))$ values by the model. The r^2_{adj} can take values in the range $(-\infty, 1]$ [24], where a value closer to 1.0 implies that the model explains the variance in the set of data used for fitting well.

V. Illustrations

To illustrate the ability of time series models to predict the number of software failures in future intervals, the ARIMA and ARIMA-W models were applied to the J2 dataset [25] composed of the number of software failures per interval. To identify the order of the models applied to the J2 dataset, the autocorrelation function (ACF) and partial autocorrelation function (PACF) [26] were used to analyze the relationship between observations at different time lags. ACF computed the correlation between the original time series and its lagged values at different lag intervals to identify the order of the moving average component, while PACF measured the correlation between the original time series and its lagged values, after removing the effects of intermediate lags to identify the order of the auto-regressive component. For the ARIMA models, The ACF and PACF were applied to $(n - v)$ data points, where $n = 181$ intervals for the J2 dataset, and ν was varied to illustrate four scenarios using 50%, 60%, 70%, and 80% of the data for model fitting. For ARIMA-W models, the ACF and PACF were applied to each window of data. For the sake of illustration, each portion of the dataset considered for model fitting was divided into four windows of equal length n_w, each containing the sequential time series and non-overlapping data points. For example, for the scenario where 50% of the data was used for model fitting, the length of the windows was $n_w = (n * 0.5)/4 = (181 * 0.5)/4 \approx 23$. This approach allowed each window to possess the necessary order of auto-regressive and moving average components to accurately capture patterns and relationships present within that specific window.

The ARIMA model development proceeded through a systematic sequence of steps. (i) Initially, time series models were constructed using the earliest correlated lagged values identified from the ACF and PACF. (ii) Subsequently, least squares estimation was applied to a subset of the data to estimate model parameters, followed by the computation of goodness-of-fit measures to validate the models. (iii) The number of lagged values for the time series variable FC was then systematically increased following the ACF and PACF lists of correlated lags. (iv) This iterative process continued until no combination of lags for the auto-regressive and moving average components could enhance the adjusted coefficient of determination r^2_{adj}. (v) This approach was carried out for 50%-80% of the data, with the remaining data serving as a validation set to assess models' performance. The application of the ARIMA-W model followed a similar process, yet with a key difference: the dataset used for model fitting was divided into four sequential windows of equal length. Consequently, these models identified lags specific to each window and independently estimated parameters for each. As discussed in Section III, predictions of future intervals were made using the model constructed in the 4^{th} window. The goodness-of-fit measures for ARIMA-W models were computed using the joint FC predictions from all four windows and the future instances.

Table I reports the order of models representing the lagged observations and forecast errors as well as the degrees of differentiation required to maximize the r^2_{adj}. In the traditional approach, a single ARIMA model was applied to the complete dataset, whereas the ARIMA incorporating the windowing method applied an ARIMA model to each window individually. Table I also shows the number of parameters ρ contained in each model, and the associate goodness-of-fit values achieved, which were penalized for the number of

TABLE I
VALIDATION OF MODELS' PREDICTION ON J2 DATASET

Data Subset for Model Fit	Method	Model	Parameters (p)	RMSE	PRMSE	r^2_{adj}
50% = 90 data points	Traditional	ARIMA(1,0,3)	5	11.9627	**32.6696**	0.9637
	Windowing	ARIMA(1,0,3) + ARIMA(1,0,2) + ARIMA(2,0,1) + ARIMA(1,0,1)	16	**5.1277**	100.3612	**0.9928**
60% = 108 data points	Traditional	ARIMA(6,0,16)	23	8.2421	**134.5442**	0.9827
	Windowing	ARIMA(11,0,11) + ARIMA(2,0,9) + ARIMA(1,0,1) + ARIMA(3,0,3)	45	**3.3541**	163.0272	**0.9966**
70% = 126 data points	Traditional	ARIMA(1,0,3)	5	9.6629	46.9954	0.9812
	Windowing	ARIMA(1,0,11) + ARIMA(2,0,2) + ARIMA(1,0,2) + ARIMA(2,0,2)	27	**5.0928**	**9.7801**	**0.9944**
80% = 144 data points	Traditional	ARIMA(7,0,2)	10	8.5367	20.4357	0.9871
	Windowing	ARIMA(3,0,3) + ARIMA(2,0,2) + ARIMA(3,0,3) + ARIMA(9,0,9)	38	**5.1627**	**2.5883**	**0.9950**

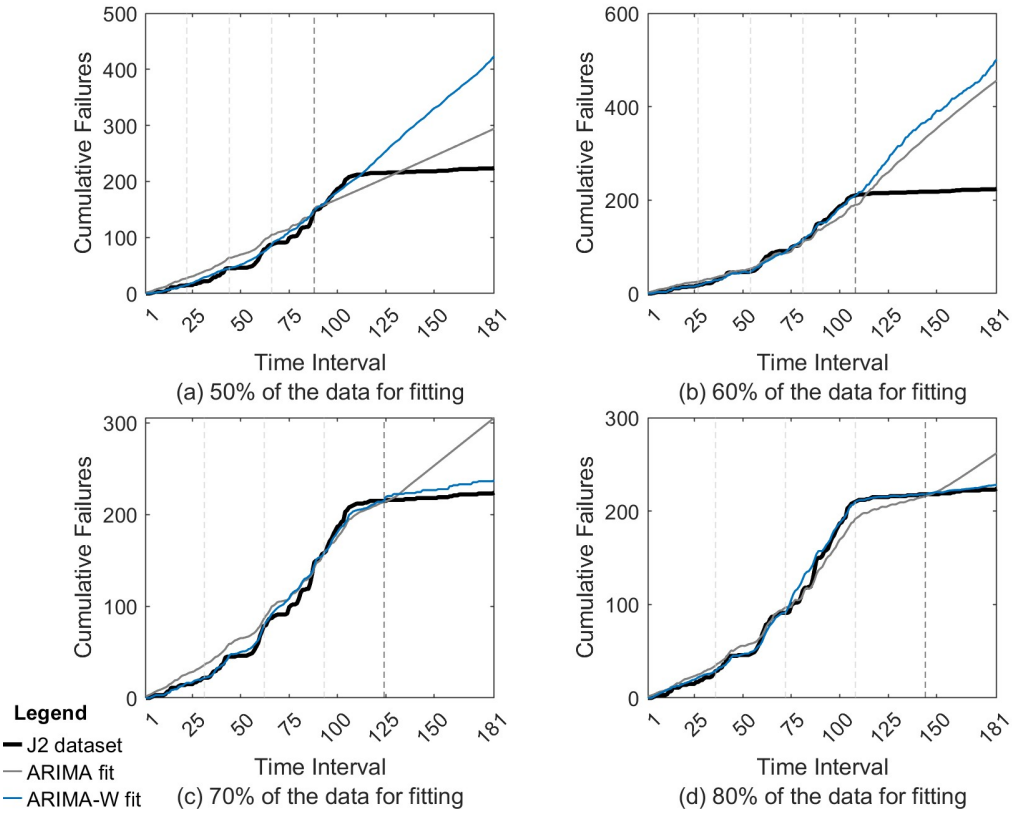

Fig. 1. ARIMA and ARIMA-W fits with first 50%-80% of the J2 dataset.

parameters in the model, to quantitatively compare ARIMA and ARIMA-W models. The model that performed best overall by exhibiting the highest r^2_{adj} and lowest $RMSE$ is highlighted in bold in Table I for each of the four subsets of data used for model fitting. Table I indicates that the ARIMA-W models exhibited superior r^2_{adj} and the best $RMSE$ in all four cases considered, suggesting effective tracking of software failure counts. However, when 50% and 60% of the data were used to fit the model, the traditional ARIMA model exhibited lower $PRMSE$. This outcome implies that the numerous parameters in the ARIMA-W models were able to adjust to minor variations in the sample used for model fitting, but struggled to discern the underlying patterns that could generalize to new instances. Nevertheless, as more data became available, the performance of ARIMA-W models improved progressively, exhibiting significant enhancements particularly in $PRMSE$, where ARIMA-W achieved $46.9954/9.7801 = 4.8$ and $20.4357/2.5883 = 7.9$ times lower $PRMSE$ than ARIMA for the 70% and 80% subsets, respectively. These results indicate that ARIMA-W models are much more precise in predicting future software failures when fitted with a larger historical dataset of software failures.

To visually compare the ARIMA and ARIMA-W model fits on each subset of data used for model fitting reported in Table I, Figure 1 shows the empirical data as well as how each model tracks and predicts, where the dashed light gray vertical

line correspond to each window considered in the ARIMA-W model, and the dashed dark gray vertical line corresponds to the data point where tracking ends and prediction begins for all models. In Figure 1-(a) it becomes apparent that when only half of the available data was employed for model fitting, the models successfully captured the general trend present in the dataset. However, they struggled to accurately replicate this trend when applied to the unseen subset of data, indicating a limitation in their ability to generalize beyond the data they were fitted on. Figure 1-(b) shows that when 60% of the data was used to fit the models, both models achieved a very notable improvement in the model fit, but still exhibited difficulty in accurately characterizing the unseen data subset. Figure 1-(c) and Figure 1-(d) show that increasing the subset of data used for model fitting to 70% or 80% considerably improved the predictive ability of the ARIMA-W models. By dividing the dataset into smaller windows and allowing the model to adapt to the nuances present within each window, the models were better equipped to handle the variations and complexities inherent in the data. Consequently, this results in a significant enhancement in their predictive accuracy, highlighting the critical role that data availability and partitioning strategies play in refining the predictive capability of models to predict software failures.

VI. Conclusion and Future Research

This paper introduced an ARIMA model incorporating a windowing technique (ARIMA-W) to track and predict the number of software failures over time. The ARIMA-W mitigates the impact of non-linearity in software failure observations, which is a challenge faced by traditional ARIMA models. To validate the proposed model, both ARIMA and ARIMA-W models were evaluated using an actual dataset composed of the number of software failures in 181 intervals, using 50%, 60%, 70% and 80% of the data for model fitting. Our results revealed that while both the ARIMA and ARIMA-W performed well with limited data, the ARIMA-W outperformed ARIMA when more data were available for model fitting, where ARIMA-W achieved approximately 5 and 8 times lower $PRMSE$ when 70% and 80% of the data were used to fit the models. Visual comparisons of model fit further supported the superiority of ARIMA-W to capture data complexities. These findings suggested the importance of data availability and partitioning strategies to refine predictive capability to forecast the number of software failures.

Future research will focus on refining the windowing technique for software failure predictions by identifying the optimal number of windows to be used. Moreover, non-equally sized windows will be considered to further enhance the predictive accuracy of the models to improve long-term predictions.

Acknowledgments

This research was supported by the National Science Foundation under Grant Number 1749635. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. R. Lyu, ed., *Handbook of Software Reliability Engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.

[2] H. Pham, *System Software Reliability*. Springer Science & Business Media, 2006.

[3] S. Ross, *Stochastic Processes*. No. v. 1 in Probability and Statistics Series, Wiley, 1983.

[4] W. Farr and O. Smith, *Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) Library Access Guide. Revision 3*. Naval Surface Warfare Center Dahlgren Division VA. - Defense Technical Information Center, 1993.

[5] U. Raja, D. P. Hale, and J. E. Hale, "Modeling software evolution defects: a time series approach," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 21, no. 1, pp. 49–71, 2008.

[6] S. Yamada and S. Osaki, "Reliability growth models for hardware and software systems based on nonhomogeneous poisson processes: A survey," *Microelectronics Reliability*, vol. 23, no. 1, pp. 91–112, 1983.

[7] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on Reliability*, vol. 32, no. 5, pp. 475–484, 1983.

[8] A. Goel, "Software reliability models: Assumptions, limitations, and applicability," *IEEE Transactions on Software Engineering*, no. 12, pp. 1411–1423, 1985.

[9] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. USA: Prentice Hall PTR, 3rd ed., 1994.

[10] T. M. Khoshgoftaar and R. M. Szabo, "Investigating arima models of software system quality," *Software Quality Journal*, vol. 4, pp. 33–48, 03 1995.

[11] S. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Computers Industrial Engineering*, vol. 35, no. 1, pp. 213–216, 1998.

[12] A. Amin, L. Grunske, and A. Colman, "An approach to software reliability prediction based on time series modeling," *Journal of Systems and Software*, vol. 86, no. 7, pp. 1923–1932, 2013.

[13] S. Aljahdali and M. El-Telbany, "Software reliability prediction using multi-objective genetic algorithm," in *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 293–300, IEEE, 2009.

[14] J.-H. Lo, "A study of applying arima and svm model to software reliability prediction," in *2011 International Conference on Uncertainty Reasoning and Knowledge Engineering*, vol. 1, pp. 141–144, IEEE, 2011.

[15] M. Goulão, N. Fonte, M. Wermelinger, and F. Brito e Abreu, "Software evolution prediction using seasonal time analysis: A comparative study," in *2012 16th European Conference on Software Maintenance and Reengineering*, pp. 213–222, 2012.

[16] K. Kumaresan and P. Ganeshkumar, "Software reliability prediction model with realistic assumption using time series (s)arima model," *J Ambient Intell Human Comput*, vol. 11, pp. 5561–5568, 2020.

[17] K. K. Raghuvanshi, A. Agarwal, K. Jain, and V. B. Singh, "A generalized prediction model for improving software reliability using time-series modelling," *International Journal of System Assurance Engineering and Management*, vol. 13, pp. 1309–1320, 06 2022.

[18] D. Alberg and M. Last, "Short-term load forecasting in smart meters with sliding window-based arima algorithms," *Vietnam Journal of Computer Science*, vol. 5, no. 3, pp. 241–249, 2018.

[19] H. Dong, X. Guo, H. Reichgelt, and R. Hu, "Predictive power of arima models in forecasting equity returns: a sliding window method," *Journal of Asset Management*, vol. 21, no. 6, pp. 549–566, 2020.

[20] S. Sheoran and S. Pasari, "Efficacy and application of the window-sliding arima for daily and weekly wind speed forecasting," *Journal of Renewable and Sustainable Energy*, vol. 14, no. 5, p. 053305, 2022.

[21] H. Pham, "Software reliability," *John Wiley & Sons*, 1999.

[22] R. D'Agostino, *Goodness-of-fit-techniques*. Routledge, 2017.

[23] A. K. Srivastava, V. K. Srivastava, and A. Ullah, "The coefficient of determination and its adjusted version in linear regression models," *Econometric Reviews*, vol. 14, no. 2, pp. 229–240, 1995.

[24] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021.

[25] Lance Fiondella's Dependable Software and Systems Lab Resources, "Example failure data sets." https://github.com/LanceFiondella/DSSL-Resources/blob/main/SFRAT/example_failure_data_sets.xlsx, 2020.

[26] Z. A. Lomnicki and S. K. Zaremba, "On the estimation of autocorrelation in time series," *The Annals of Mathematical Statistics*, vol. 28, no. 1, pp. 140–158, 1957.