Utilizing Time-Series Forecasting in Software Project Planning and Software Quality Management

Asodariya, Denish Dalsukhbhai, Illinois Institute of Technology, dasodariya@hawk.iit.edu
Vaghani, Utsav Shaileshbhai, Governors State University, uvaghani@student.govst.edu
Shah, Meet Pragneshkumar, Governors State University, mshah6@student.govst.edu

***Abstract*— Since the last decade, data science has become one of the widely known fields in the technological industry. In the realm of data, Time series forecasting is one of the toughest problems because of its accuracy and methodology. Moreover, it involves prediction for different variables for different time intervals. Due to recent artificial intelligence development time series forecasting has gained quite an attraction from the industry. There are a variety of methods used in time series forecasting and some are widely used. There are different methods for different types of data. When it comes to software management it means whether it will be a success or failure. For management and quality, we need to keep some variables in view before selecting forecasting methods like security, bug detection, bug solving, optimization, project development phase, etc. We need to find a forecasting method that covers these points. In this paper, we will see some of the common and widely used time series forecasting methods like ARIMA (Autoregressive Integrated Moving Average), LSTM (Long-Short term memory), Facebook Prophet, and GANS(Generative Adversarial Neural Networks). which will be best for software management and quality with its problem and benefits. In this paper, we will provide some real-life examples of these methods. How does it work?**

I. INTRODUCTION

ACCURANCY in forecasting is one of the important points in project planning management because it decides the success rate of the project. Project planning is one kind of roadmap for the development of the project which has different variables like timelines, resources, deliverables, processes, task allotment, etc. while quality management contains variables like tests, code reviews, bug finding, bug resolving, resource limitation, etc. there are many different types of forecasting methods, and they work based on different variables also they have their success rate. So, it is hard to find which forecasting method will be the best for project management. This is research about which time series forecasting methods are highly effective when it comes to software development and quality management.

II. LITERATURE REVIEW

A. Exploration of Existing Literature on Time-Series Forecasting in Software Project Planning and Quality Management

Software project management (SPM) and software project quality management (SPQM) have evolved to rely more on time-series forecasting since they may predict future project performance through historical data analysis. By looking for patterns and trends in past projects, project managers may make well-informed decisions about scheduling, risk reduction, allocating resources, and maintaining quality.

Research has looked at the value of time series forecasting in SPM and SPQM. According to Menzies et al. (2007), time series forecasting models outperformed expert judgment and analogy-based methods for estimating the software project effort and defect count predictions.

In a similar vein, Kitchenham et al. (2009) conducted a systematic review underlining the accuracy of time series forecasting methodologies in SPM and SPQM concerning project effort, cost, schedule, and defect count, especially for larger and more complex projects.

B. Examination of Different Time-Series Forecasting Methods and Their Applicability to Software Projects

In SPM and SPQM, a variety of time series forecasting techniques have been employed; each has advantages and disadvantages. There are methods in use:

- **ARIMA (Autoregressive Integrated Moving Average):** By considering values and the current value of a random error term, this statistical model is used to predict the value of a time series. ARIMA is adaptable. can be used for time series data elements like defect count, project effort, cost, and timeline.
- **Exponential smoothing:** This technique gives observations progressively smaller weights in an easy-to-understand manner. It is quite good at predicting time series containing trends, such as the budget and timeline of a project.
- **Neural Networks:** Robust machine learning models proficient in capturing intricate nonlinear relationships in time series data, proving highly effective in forecasting time series characterized by high noise levels, such as defect count.

The particulars of the project data and the required forecasting accuracy figure out which time-series forecasting technique is best. For example, when project data has a clear trend, exponential smoothing seems like a reasonable choice, nevertheless, very noisy project data may require the use of a neural network.

C. Illustration of Case Studies and Successful Implementation of Time-Series Forecasting in Software Project Planning and Quality Management

The effective fusion of time-series forecasting in SPM and SPQM is demonstrated by numerous case studies. In an example scenario where they used it to predict the number of defects in a large-scale software development project, Nagappan et al. (2008) showed the accuracy of time-series forecasting in predicting defect count and helping proactive steps against possible quality issues.

Ruhe (2014) made a similar estimate of the duration of software development operations using time-series forecasting. The use of time-series forecasting, which offers precise forecasts of project performance, may help project managers distribute resources more effectively, improve project quality, reduce project risks, and complete projects successfully.

These case studies demonstrate how the use of time-series forecasting may improve SPM and SPQM decision-making. Project managers may more effectively allocate resources, improve project quality, reduce project risks, and complete projects with the use of time-series forecasting, which offers precise estimates of project performance.

III. Methodology

Time series forecasting plays an important role in improving software project planning and quality management. It provides predictive insights into various projects. This paper gives detailed information about time series furcating in software development.

A. Role of Time-Series Forecasting in Software Project Planning

Time series forecasting helps in project planning by predicting risks, project timelines, and resource utilization. Project managers can create project timelines and decide how best to allocate resources by looking at past data. To improve project outcomes, this segment of the study examines how time series forecasting helps with software project strategic planning and execution.

B. Role of Time-Series Forecasting in Software Project Quality Management

In software project quality management, time series forecasting aids in predicting and mitigating potential quality issues. This subsection delves into the application of forecasting models to anticipate defects, identify code vulnerabilities, and enhance overall software quality. By integrating forecasting into quality management practices, development teams can proactively address issues, leading to higher-quality software deliverables.

C. Process of Time-Series Forecasting

In this part, the whole process used to incorporate time series forecasting into software project planning and quality control is described. Ensuring a methodical and efficient approach, the methodology covers all the important phases, from defining the issue domain to model deployment.

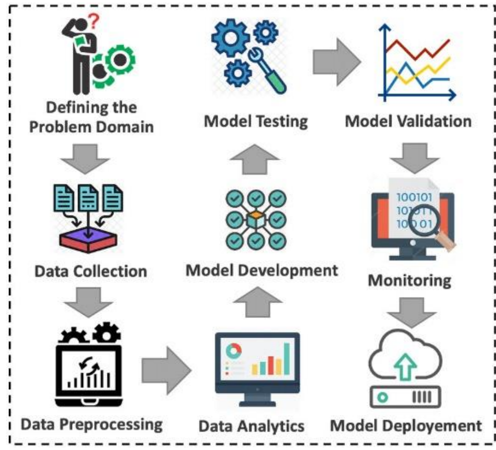

Fig 1. Process of Time-series Forecasting [10]

- **Defining The Problem Domain:** Clearly describing the time series forecasting issue domain inside the framework of a software project is the first stage in the process. This involves specifying the precise project features and goals—such as project length, resource allocation, or quality metrics—that the forecasting model is meant to address.
- **Data collection:** The Accuracy of the forecasting is dependent on the effectiveness of data collection. This part is about that. There are different kinds of data like historical project data, different timelines, use of resources, and relevant metrics. So, it is very important to use the right data collection system. We always focus on obtaining strong datasets for training and validating forecasting models.
- **Data Preprocessing:** Preprocessing is one of the very important parts of model design. This part includes structuring, cleaning, and converting data so it satisfies the needs of training. Currently, there are different types of data preprocessing techniques available like normalization, missing values handling, and feature engineering.
- **Model Testing:** Model testing is one of the important aspects of quality assurance. After training the model it is important to test it several times. There are many techniques like the fuzzy technique which uses rare inputs to check the model. It is important to check the model for different scenarios and conditions. Testing

the model with historical data can help in selecting the right model.

- **Model Development:** After choosing the right model goes into further development part to improve its performance and forecasting. This includes improving the optimization, training for fuzzy data, parameter adjustment, etc.
- **Model Validation:** It is important to check the reliability of the model in the real world. After the development of the model, this part takes some real-world problems and places them into the model. After it checks the results from the model and checks its accuracy.
- **Monitoring:** Continued monitoring of the model can reflect minor model changes. Which is very helpful in improving accuracy over time. This section includes which monitoring testing is best for the monitoring model. It also shows events and conditions which are very rare to occur and how to train a model for that and which are important changes to make. Monitoring shows provides surety that the model will stay effective.
- **Deployment:** This is the final step of software management, in which we use the model to forecast data and convert it into real-time action with software project planning. This section sheds light on the hidden bugs and errors behind the project. Overall, the deployment is making your model practical in the real world and observing its effects.

IV. TIME-SERIES FORECASTING TECHNIQUES

Planning software projects and quality control are made more effective when time-series forecasting approaches are used. The various forecasting techniques—ARIMA (Autoregressive Integrated Moving Average), LSTM (Long-Short Term Memory), Facebook Prophet, and XGBoost (Extreme Gradient Boosting)—are thoroughly examined in this section.

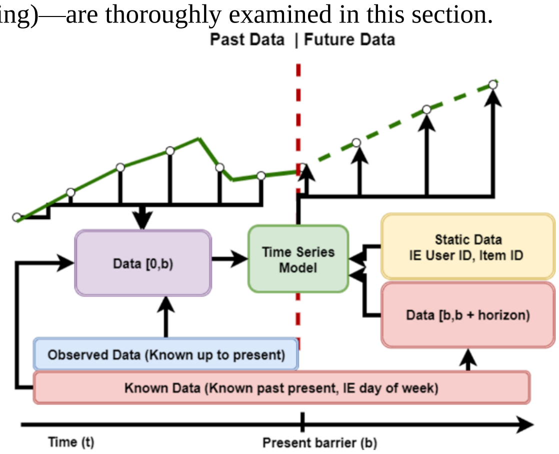

Fig 2. Time-series Forecasting Techniques [9]

A. ARIMA(Autoregressive Integrated Moving Average)

It is one of the widely used and accurate time forecasting techniques. It is one of the parts of the box Jenkins models. It has different components like Autoregressive (AR), Integrated, and Moving Average. Arima uses the parameters(p,d,g) where p stands for autoregressive part, d stands for differencing and q stands for the average part. The autoregressive model provides the connection between several aged observations and observations. As an equation, it can be expressed as:

$$Y_t=c+\phi_1 Y_{t-1}+\phi_2 Y_{t-2}+\ldots+\phi_p Y_{t-p}+\varepsilon_t$$

Here Y_t is the current observation c is constant, ϕ is an autoregressive coefficient, ε_t is noise error and Y_{t-1} is lagged observations.

The integrated component is used for differencing time series data for that the equation is $\Delta Y_t=Y_t-Y_{t-1}$. The moving average is used to find the relationship between residual error and observation. It can be expressed as:

$$Y_t=\mu+\varepsilon_t+\theta_1\varepsilon_{t-1}+\theta_2\varepsilon_{t-2}+\ldots+\theta_q\varepsilon_{t-q}$$

Here μ is the mean, εt-1 is the lagged error, θ is an average coefficient, and ε_t is the current error. The ARIMA model is mainly used to exhibit trends and seasonality. It is useful in different fields like finance and economics for performing tasks like demand and stock price forecasting. For the seasonality, we use a similar model called a seasonal ARIMA or SARIMA. It adds seasonal components as an add-on parameter.

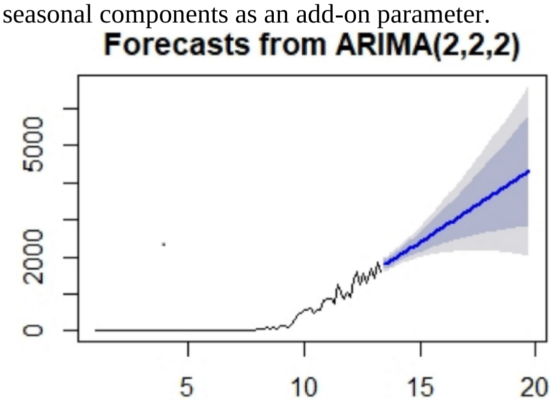

Fig 3. ARIMA Model[11]

B. LSTM(Long-Short-Term Memory)

LSTM provides a strong network for time series forecasting. It has the unique ability to capture intricate dependencies within sequential data. The initial phase called data preparation, in which time series data involves input and output sequence. For optimal performance normalize data. In this step, data has been divided into two parts training and testing.

In this process, we need to take care of the number of layers, neurons, and other parameters. Once the model is compiled, the training phase starts. It evaluates the model and guides against overfitting, unseen data, and its adaptability. The following test involves models on previously unseen data, enabling the generation of predictions for future time points.

During this dynamic process, vigilance and adaptability are important. The post-processing step is crucial to revert data transformations and for final prediction. For the practical implementation, one can do it with Python libraries such as

Keras. where we can play around with different settings to make our model work better for the specific data we have. In simple words, with the help of this technique, we are teaching computers to be good at guessing the future based on past data.

Here is the primary equation of LSTM:

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Here, i_t, f_t, g_t, and o_t are the input, forget, cell, and output gates, respectively. X_t is the input at the time. ht-1 and ct-1 are the hidden state and cell state from the previous time step. W_{ii}, W_{if}, W_{ig}, and W_{i0} are the weight matrices for the input gates. W_{hi}, W_{hf}, W_{hg}, W_{ho} are the weight matrices for the hidden state gates. b_{ii}, b_{if}, b_{ig}, and bio are the bias vectors for the input gates. b_{hi}, b_{hf}, b_{hg}, b_{ho} are the bias vectors for the hidden state gates. s is the sigmoid activation function, and tanh is the hyperbolic tangent activation function.

This equation describes how an LSTM cell processes input and updates its cell state and hidden state over time, allowing it to capture long-term dependencies in sequential data.

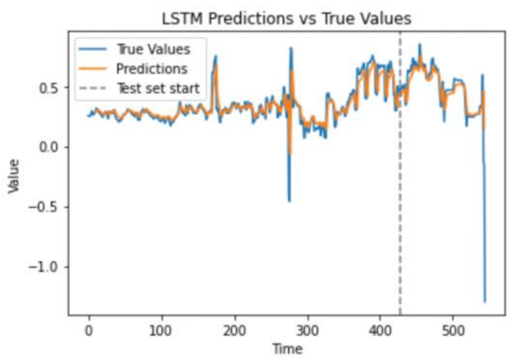

Fig 4. LSTM Model [7]

C. Facebook Prophet

The Facebook prophet forecasting technique is a product of Facebook and was developed by Fakebook's core data science team. This technique is mainly used in time-related data. When there is data that has holidays or some special occasion at a particular time in this kind of scenario Facebook Prophet is quite helpful. This technique is very useful in intricate data patterns.

A Facebook prophet can easily handle yearly and weekly. It can easily make predictions in conditions of multiple seasonal patterns. The general equation is:

$$y(t)=g(t)+s(t)+h(t)+\varepsilon_t$$

Here, y(t) is the observed value (At time t), g(t): represents the overall direction of the time, s(t) seasonal component, h(t) special events, and ε_t is the error. For trend, the function is:

$$g(t) = \sum_{i=1}^{N} \left(k_i \cdot \text{floor}\left(\frac{t-t_i}{s_i}\right)\right)$$

Here "k_i" is the i-th line segment, "t_i" is the i-th changepoint, and "s_i" is the scale parameter for the i-th changepoint.

For the seasonal component, the function is:

$$s(t) = \sum_{j=1}^{J} \left(A_j \cdot \sin\left(\frac{2\pi jt}{T_j}\right) + B_j \cdot \cos\left(\frac{2\pi jt}{T_j}\right)\right)$$

Here, A_j, and B_j are coefficients for the j-th component and T_j is the period of the j-th component.

For Holiday, the function is:

$$h(t) = \sum_{k=1}^{K} \delta_k \cdot \text{holiday}_k(t)$$

Here δ_k effect of k holiday and $\text{holiday}_k(t)$ is an indicator.

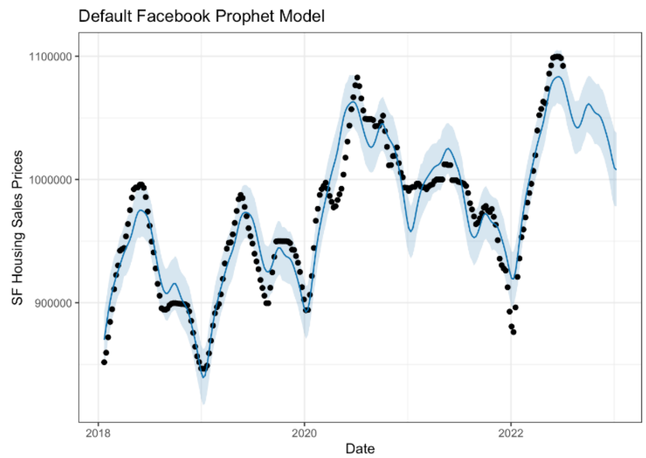

Fig 5. Facebook Prophet Model [4]

D. XGBoost (Extreme Gradient Boost)

XGBoost has one of the best predictive performances. That's why it's called a state-of-the-art machine learning algorithm. It used gradient-boosting learning techniques. It used decision trees and a series of weak learners to make errors correct which are made by preceding models. The best thing about XGBoost is its objective function. XGBoost provides enhanced efficiency.

It isn't only designed for time forecasting. It uses d for some general approaches like Data preparation, feature engineering, model training, validation, Train-Test split, and post-processing.

XGBoost is indeed used for time series forecasting but it is also true that it isn't as effective as ARIMA, SARIMA, and Facebook Prophet. However, in some situations, this method is quite helpful. Due to its versatility, it is quite famous in different domains like health care and finance. As a function for t-th boosting it can be expressed as:

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t-1)}) + \sum_{k=1}^{t} \Omega(f_k),$$

Here y_i is the true label of the i instance, $\hat{y}_i^{(t-1)}$ is the predicted value for the i-th instance, and f_k is an ensemble for the k-th tree.

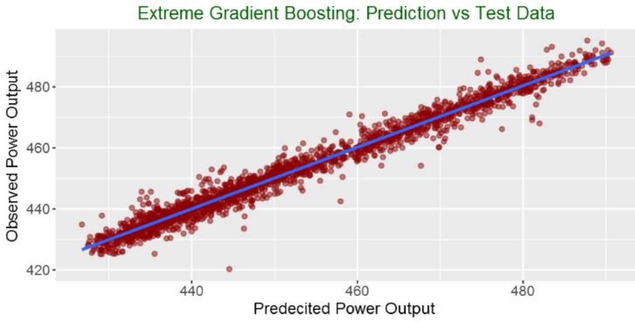

Fig 6. XGBoost Model [8]

E. Comparison of These Techniques in the Context of Software Project Planning and Quality Management

Technique	Strengths	Weaknesses	Applicability to Planning	Applicability to Quality Management
ARIMA	Captures linear trends	Limited for non-linear patterns	Yes	Limited
LSTM	Handles complex dependencies	Computational intensity	Yes	Yes
Facebook Prophet	Addresses missing values	Less effective for long-term forecasts	Yes	Limited
XGBoost	Captures non-linear relationships	Sensitivity to outliers	Yes	Yes

Fig 7. Comparison of Techniques [4]

V. Case Study

A. Time-series forecasting in Software Project Planning

One of the tech giants “UBER” is well known for using time series forecasting to improve software project planning and allocation of resources. This company heavily relies on data, so it needed the best way to optimize its data in one place time series forecasting is mainly used in predicting the demand of riders.

Every day Uber collects large amounts of data on rider demand patterns including different factors like time, day, year, events, and geographical location. This data goes into the preprocessing phase which handles missing values, outliers, and other formats that are suitable for forecasting.

Uber used traditional time series forecasting methods like ARIMA, and seasonal decomposition methods with machine learning to make forecasting. This model is trained based on historical data. Uber is constantly improving their model by providing a huge set of historical data.

Like this “Spotify” was facing issues with its large microservice architecture. To resolve this issue, they used the ARIMA time series forecasting model to predict the defect at the time of product release. which provided appreciative output with 85% accuracy in predicting defects which is used to resource allocation for Spotify to resolve defects.

Online booking giant “Booking.com” used the LSTM model for resource allocation in software development tasks using historical data. This model got 90% accuracy of prediction. Which prevented this company from overstaffing on projects.

“Netflix” is one of the online streaming giants. It has a challenge of predicting user growth and contents for that it must predict the requirement of server capacity. To resolve this problem, they used the Facebook Prophet model to predict content generation and user growth with the help of historical data. This model proved quite helpful to Netflix which leads it to efficient server allocation.

“Google” is one of the biggest tech giants in the world. It was facing an issue with predicting software development time. To predict this Google developers used the DeepSpeed machine learning model that uses historical data and dependencies on tasks to predict development time. This method provided 20% more accurate results than traditional methods, which was proven quite helpful to this company for better planning and resource allocation.

Not only tech companies are using forecasting techniques, but “Airbus” one of the airline giants used the SARIMA (Seasonal Autoregression integrated moving average) model to forecast its future maintenance cost based on historical data. This method helps the company to optimize maintenance and reduce operation costs.

B. Time-series Software Project Quality Management

“NASA” is one of the biggest space agencies in the world. The company wants to predict the defect rate in spacecraft to ensure the safety of astronauts and spacecraft. For forecasting this they used the ARIMA prediction model to analyze historical defects. This model helped to find high-potential defects which occurred in past. This became quite fruitful for the company too; they succeeded in stopping many disasters from occurring. They also resolve many software-related problems.

“Microsoft” is One of the tech giants. While releasing the Azure DevOps, the Company wanted to resolve the report as soon as possible, so they used the LSTM-based deep learning forecasting method to predict reported bugs using historical data. This model predicted high numbers of bugs and they were prepared for it. It turns out true. Due to early predictions, they were able to allocate resources and solve the bugs in minimum time.

The University of California, Irvine, researchers used time series forecasting to predict the number of code defects in open-source projects. They used past data on volume, code that is rewritten and their forecasting model predicted future defecate levels.

These approaches encourage developers to test and allocate resources potently. As a result, it improves their software quality.

Microsoft had issue with software release date, so research and development team used time series forecasting model to predict release date. They developed one model which examines past data of the company, which content development process, task completion rates, and other external components. This model predicts accurate date of software release date. As a result, company can manage risk management and reduces software release delays.

One of the biggest transport and food service company “Uber” was facing bottlenecks issue in their service platform, which leads to low efficiency in the application. Uber developer used time series forecasting with Markov models. They analyze historical data and analyze performance metrics. As a result, they found specific areas which needs optimized code and improved passenger satisfaction. It reduces operation cost and it also increase service quality of Uber.

VI. Conclusion

A. Outcomes

Integration of time-series forecasting into software project management offers many implications. Ensemble methods and advanced machine learning techniques improve the accuracy and reliability of project future prediction. To do more proactive decision making one should utilize dynamic and adaptive models which have Real-time ability and user-friendly interface. In fostering predictive analysis, it is important to consider ethical rules, addressing bias and transparency. Organizations are using this advancement of time series forecasting to enhance overall management efficiency. They are investing in training users and creating strong governance structures.

Based on research, the utilization of advanced machine learning techniques, such as deep learning and recurrent neural networks, in time-series forecasting proves beneficial to advance the field of software project management. For enhancing reliability, bagging, and boosting ensemble methods can be used. The adoption of dynamic models improves forecasting robustness. To ensure responsible predictive analysis, organizations should enhance ethical consideration, transparency, and bias mitigations.

B. Future Directions and Recommendations

In the future time series forecasting can be used in advanced machine learning techniques such as deep learning and recurrent neural networks. To increase model reliability one can, use bagging and boosting ensemble methods. Dynamic and adaptive models adjust models according to the external environment. For instance, according to market conditions, it can improve overall forecasting robustness. To give project managers updated information models should have real-time capabilities and user-friendly interfaces. For the best understanding of forecasting models’ evaluation metrics of developed models and different case studies are needed. It is very important to be fair and transparent while using predictive analytics in project management. This helps in this modern world of software projects to become more accurate and ethical.

The trainers and different organizations must start defining specific objectives and understanding data. One should choose models like ARIMA, LSTM, XGBoost, or Facebook Prophet which are aligned with the project characteristics and regularly evaluate its performance. To increase accuracy ensemble methods and for continuous improvement feedback loops can be used.

References

[1] Kalouptsoglou et al., "Time Series Forecasting of Software Vulnerabilities Using Statistical and Deep Learning Models," Electronics, vol. 11, no. 18, p. 2820, Sep. 7, 2022. https://doi.org/10.3390/electronics11182820

[2] Khandare, M. A. and Vyas, G. S., "Project Duration Forecasting Using Earned Value Method and Time Series," International Journal of Engineering and Innovative Technology (IJEIT), vol. 1, no. 4, April 2012.

[3] Aregbesola, M. K., "Software Project Planning with Tracking and Oversight," International Journal of Scientific and Research Publications, vol. 7, no. 6, June 2017, ISSN 2250-3153, www.ijsrp.org.

[4] H. R., Yatish and Swamy, S. R., "Recent Trends in Time Series Forecasting– A Survey," International Research Journal of Engineering and Technology (IRJET), vol. 07, no. 04, pp. 5623, Apr. 2020. e-ISSN: 2395-0056, p-ISSN: 2395-0072, www.irjet.net.

[5] Jamsutkar, K., Patil, V., and Chawan, P.M., "Software Project Quality Management," International Journal of Engineering Research and Applications (IJERA), vol. 2, no. 3, pp. 686-690, May-Jun 2012. ISSN: 2248-9622, www.ijera.com.

[6] Mahdi, M. N., et al., "Software Project Management Using Machine Learning Technique—A Review," Applied Sciences, vol. 11, no. 11, p. 5183, 2021. Available: https://doi.org/10.3390/app11115183.

[7] Rowe, M., "Using PyTorch to Train an LSTM Forecasting Model," Medium, [Online]. Available: https://medium.com/@michaelrowe/using-pytorch-to-train-an-lstm-forecasting-model-e5a04b6e0e67. Accessed: December 1, 2023.

[8] Berhane, F., "Extreme Gradient Boosting with R," R Project, Mar. 5, 2018. [Online]. Available: https://datascienceplus.com/extreme-gradient-boosting-with-r/. Accessed: December 1, 2023.

[9] Kyle Kranen, Pawel Morkisz and Carl (Izzy) Putterman, “Time Series Forecasting with the NVIDIA Time Series Prediction Platform and Triton Inference Server, Feb 15, 2022. Available: https://developer.nvidia.com/blog/time-series-forecasting-with-the-nvidia-time-series-prediction-platform-and-triton-inference-server/. Accessed: December 1, 2023.

[10] Ola Surakhi, “Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm”, October 15, 2021. Available: https://www.mdpi.com/2079-9292/10/20/2518. Accessed: December 1, 2023.

[11] Pravat Kumar Shit, “Spatial prediction of COVID-19 epidemic using ARIMA techniques in India”, July 2020. Available: https://www.researchgate.net/publication/342945021_Spatial_prediction_of_COVID-19_epidemic_using_ARIMA_techniques_in_India. Accessed: December 1, 2023.