Estimating Task Completion Times for Network Rollouts using Statistical Models within Partitioning-based Regression Methods

Venkatachalam Natchiappan*, Shrihari Vasudevan* and Thalanayar Muthukumar†
Ericsson, *India and †USA
{n.natchiappan|shrihari.vasudevan|thalanayar.muthukumar}@ericsson.com

Abstract—This paper proposes a data and Machine Learning-based forecasting solution for the Telecommunications network-rollout planning problem. Milestone completion-time estimation is crucial to network-rollout planning; accurate estimates enable better crew utilisation and optimised cost of materials and logistics. Using historical data of milestone completion times, a model needs to incorporate domain knowledge, handle noise and yet be interpretable to project managers. This paper proposes partition-based regression models that incorporate data-driven statistical models within each partition, as a solution to the problem. Benchmarking experiments demonstrate that the proposed approach obtains competitive to better performance, at a small fraction of the model complexity of the best alternative approach based on Gradient Boosting. Experiments also demonstrate that the proposed approach is effective for both short and long-range forecasts. The proposed idea is applicable in any context requiring time-series regression with noisy and attributed data.

Index Terms—Network-rollout planning, Forecasting, Tree/partition-based regression

I. INTRODUCTION

Mobile telecommunications technology is continuously evolving. The quest for supporting greater multimedia content at increased speed and scale (more consumers and devices) has resulted in significant advancements in both hardware (e.g., radio base stations, antennas and related technologies) and software (e.g., signal processing, effective use of different spectrum bands, service prioritization, etc.). These advancements and increased consumption of telecommunications services fuel greenfield (i.e., new) and brownfield (e.g., hardware upgrade) deployments require network-rollout activities. This paper focuses on developing predictive models that would enable effective planning for network-rollouts.

Communications service providers (CSPs) provide telecommunications services (e.g., messaging, telephony) to consumers; vendors build or manage the networks for the CSPs. Network rollouts occur in stages or phases, with each phase involving multiple milestones. Exemplar phases in typical network-rollouts include solution design, site acquisition, site engineering, civil works, installation, commissioning, configuration, integration, and acceptance. In any project, a delay in one of the milestones will impact the completion time of every milestone that follows it. The time duration between two milestones in a project is sensitive to multiple external factors, not all of which are controllable by project managers. A few examples of such factors are weather conditions, unforeseen unavailability of site-engineers or equipment and delays due to regulatory controls for unpredictable situations such as birds nesting on the tower. These factors need to be accounted for while allocating people and equipment for a network rollout. A tool that enables project managers to optimally plan resource utilization can enable significant cost savings for the Telecommunications industry.

Each network rollout project contains a certain set of phases based on the nature of the project; these may involve radio access network expansion, rollout of optical-fiber networks, upgrading networks - e.g., 3G to 4G, and the optimisation of network parameters to fix quality issues. The phases and consequent milestones within any project are dependent on the specific contract between a CSP and a vendor. Thus, every project at every site has attributes characterizing its metadata.

Traditionally, project managers have target dates (dictated by the contract) for the completion of each milestone. Project managers track these using an Excel spreadsheet, for a future time-period. Due to a high volume of deployment projects simultaneously executed by vendors, project managers are only able to track milestone completions occurring in a limited future period. Hence, this manual approach is highly tedious and affords only limited forward planning capabilities. This paper proposes a data and Machine Learning (ML) solution, that provides accurate short and long-range forecasts, to enable effective project planning.

Data-sets in such contexts are thus composed of attributed time-series data. Several effective first-line approaches exist for time-series forecasting; for instance, ARIMA, SARIMA, Holt-Winters [1] etc. are time-tested methods. ARIMA and its variants cannot natively handle attributed time-series data. For regression problems (including forecasting) that involve attributed data, Gradient Boosting (GB) variants (e.g., XGBoost [2]) and Random Forests (RF) are well-established approaches. The former are particularly effective due to power of additive modeling. Most classical time-series forecasting methods are statistical approaches, that make use of the statistical (e.g., linear) relationship between predictors and target variable whereas GB, RF and other such partition-based approaches are piece-wise constant in the leaves of different stumps (base estimators). Our proposed approach combines the best from both these approaches. Tree-based approaches are used to
effectively partition attributed time-series data and at the leaves, time-series regression models are employed to leverage patterns in the data. Standard ML libraries (e.g. Scikit-Learn) do not offer statistically-enhanced data-partitioned regression algorithms currently.

The contribution of this paper is to develop a forecasting solution that would enable a Telecommunications network rollout planning application. The proposed approach incorporates data trends and domain expectations to build competitive to better models, at marginal model complexity, relative to the best known approaches for the data. The approach is generic and possess significant application potential in any application context that involves attributed time-series data.

![Example Gantt chart for a network rollout project](image_url)

**Fig. 1.** Example Gantt chart for a network rollout project, depicting the phases’ timelines. The typical overlap between the phases is shown. The Build phase is a superset for activities like site engineering, civil works and installation phases.

**II. PROBLEM DEFINITION AND CHALLENGES**

The problem addressed in this paper is that of estimating the time to complete a target milestone from a source milestone, given the site’s metadata, project’s metadata, and completion time of previously completed milestones, also referred to here as intermediate milestones. Each milestone in a phase may have a dependency on a few other milestones across different phases; hence, it can be performed only after the successful completion of all dependent milestones. Certain milestones within a phase occur simultaneously, and there are overlaps between the phases as well. Also, activities across phases may be carried out simultaneously, when all of their dependent milestones were completed. Figure 1, shows an example Gantt chart with representative phases. Vertical lines represent the days to complete each phase, relative to the start date. Overlapping phases may be observed between for instance “Supply and Logistics” and “Build” phases or the “Build” and “Integration and Acceptance” phases. The time taken for each subsequent milestone is dependent on the time it took for previous milestones. In Figure 1, the time to complete a milestone in the “Supply and Logistics” phase depends on the completion times of its preceding milestones, occurring in the “Solution Design” and “Site Acquisition” phases.

A project manager maintains a “rule-of-thumb” estimation on the time duration between a target milestone and a source milestone. Such estimates are typically additive. This domain understanding suggests that a linear (regression) model may be effective at forecasting time to completion. A network rollout requires effective planning of crew, materials, and processes for achieving the numerous milestones with minimal delays and costs. Two key data-challenges encountered are -

1) Noise in data: Due to differences in operating practices, project sites exhibit different standards of data reporting, integrity and level-of-detail. This results in different levels of completeness and accuracy of data across project sites.

2) Long-tail distribution: Noise in the data is not easily differentiable. There may be genuine delays induced by bad weather conditions, material shortages, or regulatory delays stemming from a bird’s nest in the cell tower.

Standard outlier detection methods cannot differentiate between a genuine delay and inaccurate data. As a consequence, the approach proposed in this paper needs to deal with incomplete and noisy data; uncertainty in data needs to be treated at-least implicitly i.e., providing prediction bounds instead of point-predictions. Section III presents our proposed approach that overcomes the aforementioned domain and data challenges, while providing accurate predictions and uncertainty-bounds of duration to milestone-completion over different time-horizons.

**III. APPROACH**

To develop a forecasting approach given attributed time-series data, the approach proposed in this paper fuses partitioning-based approaches, such as Decision Trees (DTs), ensemble models based on it - Random Forests (RFs) and Gradient Boosting (GB) and clustering/neighborhood-based approaches, with statistical approaches. Data are partitioned based on categorical attributes and regression models are developed within each data-partition.

Linear models for regression were chosen on the basis of both domain-understanding (see Section II) and exploratory data analysis analyzing correlations between time to complete individual milestones. In other problem contexts, depending on data characteristics and modeling needs, more complex regression models like Support Vector Machines, Neural Networks, etc. may also be used. However, in such cases, the model’s explain-ability may impact its practical utilization. Also, due to generally limited training data-set sizes being available for such problem contexts and the preference for simpler, yet effective and domain-inspired models, Deep Learning methods (e.g., based on LSTMs) were not pursued within this study.

For data that conforms to a linear model, Ordinary Least Squares regression (OLS) provides a Best Linear Unbiased Estimator [3], as long as the underlying assumptions are not violated. The data used in this study exhibited both multicollinearity and heteroscedasticity. A delay in completion of any intermediate milestones will impact the prediction for the target milestone, and this cascading effect must be factored in.

---

2https://scikit-learn.org/
the model. Hence, feature elimination was not an option. Ridge Regression (RR) was tested instead of OLS to mitigate the effects of multicollinearity. To further handle data exhibiting heteroscedasticity and provide predictive-bounds instead of point-estimates, Quantile Regression (QR) was adopted as the regression method of choice, in the proposed approach. Given the earlier discussion on data being noisy, median-based metrics were preferred (over mean-based metrics) both in the cost functions of modeling approaches and in performance metrics for their evaluation, to reduce the sensitivity of the proposed approach and its performance metrics to outlier data.

Consider that data is split into $k$ partitions, which are totally exhaustive. Let $P_i$ represent a $i^{th}$ partition, created by the partitioning algorithm and $f^\alpha_i$, a statistical function that was trained using the data points belonging to the partition $P_i$ $\{(x_i, y_i)\}$ and can predict the $\alpha^{th}$-quantile value for a given $x$. The cumulative density function of the prediction/target, for a given input (test) datum, $x'$ is given by the following equation.

$$Q_\alpha(x'|f_\alpha(x_i, y_i)) = \sum_{i=1}^{k} \{1_{\{x'\in P_i\}} \cdot f^\alpha_i(x') \} \quad (1)$$

Two types of data-partitioning strategies were considered in this paper. These may be understood as being “agglomerative” or “discriminative” in nature.

1) Agglomerative partitioning methods identify subsets of data based on similarity in input-space. Methods used in this paper include clustering and k-nearest neighbors (K-NN). The k-NN approach is a limiting case of agglomerative partitioning in that it would create potentially overlapping partitions based on similarity in input-space. A data-structure (e.g., KD-tree) that facilitates nearest-neighbor search is developed using the training data. During inference, the k-nearest training data for each test point are identified. A prediction is obtained using a Quantile Regression model learned from the identified nearest-neighbor data points.

2) Discriminative partitioning methods strive for homogeneity relative to target attribute. The partitioning feature would minimize a prediction metric on the target (e.g., squared loss). Tree based approaches (e.g., Decision Trees) use this approach to perform data-partitioning.

Intuitively, the agglomerative approach may produce data-partitions that are more representative of the whole dataset, capturing the spread in the target attribute, may be less prone to over-fitting, subject to the model and its hyper-parameters and may be suited for producing predictive bounds. The latter approach, in contrast, appears to be more suited for middle-point (mean/median prediction) estimation.

A summary of the models proposed and their relationships, is depicted in Table I. Partitioning features refers to the type of features used for partitioning the dataset. These models are benchmarked with the ensemble models such as Random forest regression, Gradient boosting regression and Quantile Random Forest to understand whether using statistical models within partition-based regression models show competitive performance or not.

IV. RELATED WORK

The earliest work in the decision tree for regression tasks was addressed by [4]. The idea was to keep partitioning the feature space while there was a reduction in the loss (a parent node’s loss metric was greater than sum of the loss metric of its children). The loss function is sum of the squared deviations from the mean.

The first Piecewise-linear Decision Tree (DT) algorithm - M5 [5] was proposed by Quinlan. Initially, the algorithm grows a sufficiently deep (standard) unpruned DT then replaces subtree under non-leaf nodes with Linear Regression models based on expected error reduction. Wang and Witten extended Quinlan’s M5 model tree and named it M5’ [6]. In this extended version, the handling of missing values and categorical attributes were integrated on top of the M5 algorithm.

In [7], Loh studied different methods for splitting condition within the nodes in a tree. The paper demonstrated that it would be beneficial to fit a linear model instead of a single feature split when input features are correlated with target variable and proposed an algorithm named GUIDE, which uses linear models within DT. The approach proposed in this paper is similar to the partitioned models using Linear Regression such as M5, M5’, and GUIDE. However, our approach uses models and metrics designed to cope with noisy time-series data, combining categorical attributes and numeric features.

A non-parametric approach to estimating the entire conditional distribution of the target variable was described by [8]. Our approach differs from Quantile Regression Forest (QRF) [8] in the method of calculating quantiles, as shown by Equation 1. Our approach uses QR models inside each leaf node, whereas QRF computes the quantiles empirically, across the DTs.

Recently, a method to estimate incident duration using hybrid Decision Trees based on Quantile Regression [9] was proposed in [10]. The GUIDE algorithm (Decision Trees) based on incident characteristics was used to partition the input space. Then, a QR model was used in each leaf of the Decision Tree using traffic variables as predictors. The cited work [10] and the models discussed in our paper share a similarity in

| Models          | Partitioning type | Partitioning algorithm | Partitioning features | Base estimator |
|-----------------|-------------------|------------------------|-----------------------|---------------|
| Quantile Tree   | DISC              | Decision Tree (CART)   | All                   | QR            |
| Piecewise QR    | AGGL              | Clustering (K-Means)   | CAT                   | QR            |
| Piecewise RR    | AGGL              | Clustering (K-Means)   | CAT                   | RR            |
| Nearest Neighbor| AGGL              | Nearest Neighbor (KD-tree) | CAT               | QR            |
using the Quantile Regression model inside each partition, but our approach uses DT based on Classification and Regression Trees (CART) algorithm [11] instead of GUIDE algorithm.

V. EXPERIMENTS AND RESULTS

A. Data description

The data comprised of the status of different projects being executed across several sites. Each site was represented by categorical attributes such as city, state, region, market categories, and latitude and longitude information. Few sites had one or more projects running at the same time. Similarly, metadata information for projects included nature of the project, technology being deployed, and a list of phases involved in that particular project. The data collected on completed projects include start and end dates for every milestone in the respective projects. There were 2066 projects which contained both the source and target milestones of interest. Each project had target (planned) and actual completion times for the achieved milestones, depending on the extent of completion of the project. Figure 2 shows a histogram of the unprocessed time difference between the completion time of two milestones in the Build phase. The data exhibited a long-tail (right-skewed) distribution, showcasing the challenges mentioned in the Section II. Data were pre-processed prior to modeling and evaluation - details are provided in Appendix of [12].

![Fig. 2. A histogram on number of days to complete build-phase-milestone-1 from build-phase-milestone-2. Major portion of the projects had the duration within 100 days.](image)

B. Models evaluated

For performance evaluation, ten models were considered, chosen in accordance with section III. The approaches defined are described in the Table I. The best hyper-parameters for each models were chosen based on a five-fold cross-validation on the entire dataset. The models were fit with the best performing hyper-parameter combination on the entire training dataset. The hyper-parameters that were tuned across models include:

1) Regularization penalty values for RR, QR, Quantile Trees (QT), Piecewise linear models, and Nearest Neighbor QR.
2) Minimum number of samples to split and maximum depth of the tree for DT and RF Regression
3) Number of estimators and learning rate for Gradient Boosting models
4) Number of clusters for Piecewise linear models

C. Evaluation and results

The experiments performed sought to evaluate the performance of the proposed partitioned regression models and compare them with statistical models or tree-based ensemble models. In this paper, we show three experiments over significantly different time-frames. The first experiment has start and end milestones within the same phase (specifically, the Build phase), but the duration between the two milestones ranged up-to 120 days. The second experiment contains the start and end milestones from different phases - modelling the time duration from a site-acquisition-phase milestone to build-phase milestone. This experiment was considered for long-range forecasting (up to 350 days). For the third experiment, two milestones were chosen from the same phase (Build), which would be performed relatively close to each-other (around 60 days). This experiment tests the proposed approach in the context of short-range forecasting.

| Approach                        | Median AE (in days) | Mean AE (in days) | Number of parameters |
|---------------------------------|--------------------|-------------------|---------------------|
| Ridge Regressor                 | 6.79               | 9.55              | 51                  |
| Quantile Regressor              | 5.43               | 9.18              | 51                  |
| Decision Tree Regressor         | 7.4                | 11.12             | 20                  |
| Random Forest Regressor         | 8.19               | 11.62             | 2000                |
| QRF                             | 8.14               | 11.71             | 2000                |
| Gradient Boosting Regressor     | 4.85               | 8.35              | 2000                |
| Quantile Tree                   | 5                  | 9.69              | 140                 |
| Piecewise QR                    | 5.23               | 8.97              | 400                 |
| Piecewise RR                    | 7.08               | 9.44              | 400                 |
| Nearest Neighbor QR             | 5.07               | 9.05              | NA                  |

| Approach                        | Median AE (in days) | Mean AE (in days) | Number of parameters |
|---------------------------------|--------------------|-------------------|---------------------|
| Ridge Regressor                 | 10.04              | 14.47             | 52                  |
| Quantile Regressor              | 5.81               | 11.9              | 52                  |
| Decision Tree Regressor         | 9.3                | 14.63             | 20                  |
| Random Forest Regressor         | 9.29               | 14.57             | 2000                |
| QRF                             | 9.04               | 14.52             | 2000                |
| Gradient Boosting Regressor     | 4.89               | 9.13              | 2000                |
| Quantile Tree                   | 0                  | 5.5               | 160                 |
| Piecewise QR                    | 5.26               | 11.74             | 408                 |
| Piecewise RR                    | 9.62               | 14.28             | 408                 |
| Nearest Neighbor QR             | 5.71               | 11.51             | NA                  |

Tables II, III, and IV summarize the five-fold cross-validation performance of the models benchmarked against each-other after hyper-parameter tuning, on both median and mean absolute error (AE) metrics. The median AE metric is our primary evaluation metric due to the noise in the data but mean AE was recorded for completeness.
Simple statistical models such as Ridge Regression and Quantile Regression provided better performance than the tree based regression models because statistical models are more effective at modeling the linear relationship between the predictors and the target variable. For instance, statistical models can easily incorporate the aspect of cascading effect of delays in intermediate milestones. QT uses a one single decision tree of depth eight, whereas the tree ensemble models utilize 100 trees, each of depth 4-5, to achieve competitive performance. The drastic difference in the model complexity between the QT and Gradient Boosting regressor is evident from the number of parameters trained (140 to 2000) as part of their modelling. From all three experiments, QT was observed as one of the top three performing models, while having among the least model complexities. The table III shows that QT achieved top performance in the long range forecasting experiment with zero median absolute error using a single DT enhanced with QR models.

In short-range forecasting, where the number of intermediate milestones were fewer, all the models did comparably well. The advantage of incorporating linear models within partitioned regressors was reduced, due to the lesser number of numerical variables. In long-range forecasts, where the number of intermediate milestones was higher, linear models within partitioned regressors performed better than ensemble models. The proposed models were able to provide competitive (within ≈0.5 day median error over a 60 or 120 days prediction range, relative to the next best approach) to better (a ≈5 days reduction in median error over a 350 days prediction range, compared to the next best approach) prediction accuracy for both long-range and short-range forecasts, at a small fraction of the model complexity of the next best approach, in each case. The forecast error was within 4 to 5 days because linear models were more effective due to more intermediate milestones.

To obtain confidence bounds, the 5th and 95th percentiles were used in QR to compute the lower and upper bounds. The predicted bounds contained the true value of the target attribute (Time to complete the target milestone) in 88.72% of the projects (accuracy of the bound) and Figure 3 depicts it.

VI. CONCLUSION

This paper presented a data and ML-driven approach to predict time-to-complete milestones for the Telecommunications network-rollout planning problem. The problem involved modeling attributed time-series data, and the paper proposed statistical models within partitioned regression methods as an effective solution to the forecasting problem. The outcomes of this paper demonstrate that statistical models within data partitioning-based regression methods effectively captures data trends, tends to be at least as performant as the best-known approaches, at a small fraction of their model complexity. Leveraging data-trends can enable the simplest models to be competitive or better than sophisticated general-purpose approaches. A solution deployed based on the proposed approach provided compelling prediction errors for both short-range and long-range forecasts, suggesting that the proposed approach would be viable for enabling subsequent network-rollout planning.

REFERENCES

[1] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice. OTexts, 2018.
[2] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[3] R. A. Johnson, D. W. Wichern modeling attributed time-series data, and a proposal,” Journal of the American statistical association, vol. 58, no. 302, pp. 415–434, 1963.
[4] J. N. Morgan and J. A. Sonquist, “Problems in the analysis of survey data, and a proposal,” Journal of the American statistical association, vol. 58, no. 302, pp. 415–434, 1963.
[5] R. Koenker and K. F. Hallock, “Quantile regression,” Journal of economic perspectives, vol. 15, no. 4, pp. 143–156, 2001.
[6] K. F. Hallock, “Quantile regression forests,” Journal of Machine Learning Research, vol. 7, no. 6, 2006.
[7] W. Y. Loh, “Regression trees with unbiased variable selection and interaction detection,” Statistica sinica, pp. 361–386, 2002.
[8] N. Meinshausen and G. Ridgeway, “Quantile regression forests,” Journal of the American statistical association, vol. 58, no. 302, pp. 415–434, 1963.
[9] R. Koenker and K. F. Hallock, “Quantile regression,” Journal of economic perspectives, vol. 15, no. 4, pp. 143–156, 2001.
[10] Q. He, Y. Kamarianakis, K. Jintanakul, and L. Wynter, “Incident duration prediction with hybrid tree-based quantile regression,” in Advances in dynamic network modeling in complex transportation systems. Springer, 2013, pp. 287–305.
[11] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and regression trees. belmont, ca: Wadsworth,” International Group, vol. 432, no. 151-166, p. 9, 1984.
[12] V. Natchiappan, S. Vasudevan, and T. Muthukumar, “Estimating task completion times for network rollouts using statistical models within partitioning-based regression methods,” arXiv preprint arXiv:2211.10866, 2022.