8-2023

Forecasting Workforce Requirement for State Transportation Agencies: A Machine Learning Approach

Adedolapo Mojed Ogungbire
University of Arkansas, Fayetteville

Forecasting Workforce Requirement for State Transportation Agencies: A Machine Learning Approach

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Civil Engineering

by

Adedolapo Ogungbire
Osun state University, Nigeria
Bachelor of Engineering in Civil Engineering, 2018

August 2023
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Suman Mitra, Ph.D.
Thesis Director

_____
Sarah Hernandez, Ph.D.
Committee Member

_____
Lakeshmi Sasidharan, Ph.D.
Committee Member

ABSTRACT

A decline in the number of construction engineers and inspectors available at State Transportation Agencies (STAs) to manage the ever-increasing lane miles has emphasized the importance of workforce planning in this sector. One of the crucial aspects of workforce planning involves forecasting the required workforce for any industry or agency. This thesis developed machine learning models to estimate the person-hour requirements of STAs at the agency and project levels. The Arkansas Department of Transportation (ARDOT) was used as a case study, using its employee data between 2012 and 2021. At the project level, machine learning regressors ranging from linear, tree ensembles, kernel-based, and neural network-based models were developed. At the agency level, a classic time series modeling approach, as well as neural networks-based models, were developed to forecast the monthly person-hour requirements of the agency. Parametric and non-parametric tests were employed in comparing the models across both levels. The results indicated a high performance from the random forest regressor, a tree ensemble with bagging, which recorded an average R-squared value of 0.91. The one-dimensional convolutional neural network model was the most effective model for forecasting the monthly person requirements at the agency level. It recorded an average RMSE of 4,500 person-hours monthly over short-range forecasting and an average of 5,000 person-hours monthly over long-range forecasting. These findings underscore the capability of machine learning models to provide more accurate workforce demand forecasts for STAs and the construction industry. This enhanced accuracy in workforce planning will contribute to improved resource allocation and management.

ACKNOWLEDGEMENTS

I would like to acknowledge the role and support of everyone who were instrumental to the completion of my master thesis. First, my immense gratitude goes to my advisor, Dr. Suman Mitra for his invaluable advice and continuous support in my academic career. I am deeply grateful to my committee members, Dr. Sarah Hernandez and Dr. Lakeshmi Sasidharan for their roles and unwavering support, invaluable comments and suggestions and a memorable thesis defense. I also want to graciously acknowledge the Arkansas Department of Transportation (ARDOT) for providing access to the data utilized in this study.

To my colleagues at ZeRo lab, I would like to thank them for their support and encouragement in every step of this journey. You all have made my research experience here a fun and memorable one for me. I will also like to appreciate my friends and family for their continuous support and understanding through the period of my research and thesis writing. Without the support and love from you guys, this wouldn't have been possible.

On a final note, I would like to give thanks to the Almighty for strengthening me through this period and giving me the grace to complete my research work for a successful completion of my degree requirements.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1. Introduction

State transportation agencies (STAs) are currently grappling with the challenges of meeting their workforce requirements. In recent years, the maintenance and management of an increasing number of state-managed lanes have been entrusted to a dwindling number of engineers and inspectors within STAs (Taylor & Maloney, 2013). To address this issue, STAs must engage in effective workforce planning, which heavily relies on the ability to accurately forecast future workforce needs. The adequate planning of STA workforce is crucial for ensuring cost-effectiveness, quality, and safety in construction projects.

Previous studies on forecasting the construction engineering workforce have predominantly focused on econometrics approaches such as vector error correction, Box-Jenkins Autoregressive Integrated Moving Average (ARIMA), labor multiplier approach, and simple linear regression models. There have been limited studies on the strength of machine learning algorithms in predicting the workforce needed. This study uses the Arkansas Department of Transportation (ARDOT) case study to investigate the applicability of machine learning regressors and neural network-based time series models in predicting the workforce requirements for the agency. Workforce forecasting is carried out at different levels, which include international, national, industrial, regional, agency, and project levels (Laslett R. E., 1972). This thesis primarily focuses on forecasting the workforce at the project and agency levels, which refer to the workforce needed for individual construction projects within a state agency and the overall workforce required for the state agency to effectively carry out all its projects.

Previous efforts in employing advanced machine learning techniques in this area have been limited, with only a few research publications considering these methods (Choudhury et al., 2002). This study makes several notable contributions to the existing literature. Firstly, it

evaluates the performance of machine learning regression models in predicting person-hour requirements at the project level, exploring their differences in terms of accuracy, model training time, and prediction time. Secondly, it develops neural network-based time series models for agency-level forecasting of person-hour requirements. The time series models are also tested for their ability to forecast further into the future by examining their longer-range forecasting capabilities.

This thesis consists of five sections: the introduction is followed by a detailed literature review on the previous studies based on past methodologies that have been explored in forecasting workforce requirements across the construction industry and state transportation agencies. Then, the data and methodology of the study are presented in the third section. The fourth section describes the results of the forecasts from the machine learning models. Finally, the thesis concludes with a summary of some of the significant findings, limitations, and directions for future research.

Chapter 2. Literature Review

Workforce forecasting and planning have a rich historical background dating back to the 1940s, with evidence of its significance in military, industrial, and civilian contexts (Pate, 1943). Its practical application has spanned various industries, including medicine, construction, technology, and aviation (Azimi Nayebi et al., 2019; Benitez et al., 2013; Chung et al., 2010; Park et al., 2008). Accurate prediction of workforce requirements has yielded numerous benefits across industries, such as enhanced risk management, productivity, and cost-efficiency (Collings & Mellahi, 2009; Garza et al., 2013). Prior research has synthesized reviews on general workforce forecasting methods (Edwards, 1983; Safarishahrbijari, 2018), as well as industry- or occupation-specific approaches (Lomas et al., 1985; Prescott P. A., 1991; J. M. W. Wong et al., 2004; Y. Zhao et al., 2022). Notably, the most recent review by Y. Zhao et al., 2022, focused on the construction industry, critically examining previous forecasting methods and addressing their limitations. A significant finding of their study highlighted the unsuitability of many forecasting methods in scenarios involving sudden changes in workforce requirements. Additionally, this review emphasized Hong Kong as the leading contributor to the literature in this field, accounting for over 60% of the research efforts (Y. Zhao et al., 2022). Building upon these studies, the forecasting methods can be broadly classified into qualitative and quantitative models (Safarishahrbijari, 2018; Y. Zhao et al., 2022), which are further discussed in this section.

2.1 Qualitative Models

Qualitative models, which are valuable in situations where empirical data is lacking, and are widely employed by human resources personnel, as demonstrated in previous studies (Hagopian et al., 2012; Lagarde & Blaauw, 2009; Shemin et al., 2002). Among the various qualitative models available, the Delphi method stands out as one of the most popular and

suitable for the construction industry (Safarishahrbijari, 2018; Y. Zhao et al., 2022). The Delphi method involves engaging a panel of experts to provide responses to a series of questionnaires. Through multiple rounds of questionnaire iterations, a consensus is reached among the participants (Bryant et al., 1973). In cases where unknown parameters are a consideration, the Delphi method has been combined with other models (Safarishahrbijari, 2018). Based on the original Delphi method, several other forms of Delphi models have been proposed, which include the policy Delphi and modified Delphi, among others (Dalkey & Helmer, 1963; Gordon & Helmer, 1964; Stitt-Gohdes & Crews, 2004). A modified Delphi method was developed to forecast long-term workforce needs in the tourism industry, where industry experts ranked the importance of factors affecting the workforce based on existing literature (Vázquez-Ramos et al., 2007). Similarly, Kwak et al., (1997) utilized the Delphi technique to identify factors impacting workforce demand and supply in urban academic health centers, and further assigned importance to these factors using the analytic hierarchy process (AHP) model, considering their criticality.

Although the Delphi method offers advantages when certain factors are difficult to quantify or when empirical data is lacking, it does have limitations. As the method relies on subjective responses from a panel of experts, and often involves multiple rounds of questionnaires, it is prone to bias and may not always yield accurate results (Hsu & Sandford, 2007; Linstone & Turoff, 1975; Parker & Caine, 1996; Y. Zhao et al., 2022).

2.2 Quantitative Models

Quantitative models, which encompass mathematical representations of systems or processes, constitute the majority of methods employed for forecasting workforce needs (Safarishahrbijari, 2018). These models can be further classified into distinct groups, each characterized by its unique strengths and weaknesses.

2.2.1 Time Series Models

Time series models are statistical models used to predict future values based on the analysis of a sequence of data collected at equally spaced points in time (Box et al., 2008; Martin et al., 2012). These models rely exclusively on historical data to estimate future trends and make forecasts. Various time series methods have been applied to forecast workforce requirements in the construction industry, including vector error correction (VEC), Box-Jenkins, exponential smoothing, and the Markov processes (Y. Zhao et al., 2022). For instance, J. M. W. Wong et al., (2011) presented the performance of VEC in estimating the short to medium-term construction workforce demand. Although the VEC was found to be inferior to the log-linear regression approach, it outperformed the Box-Jenkins approach, highlighting its strength in capturing the causal relationship between workforce demand and associated factors in the construction industry. Other time series approaches, such as exponential smoothing, have also been presented and compared with the VEC and log-linear regression (J. M. Wong et al., 2009). However, these time series models have limitations as they rely solely on past data, which might only be sufficient for short-term forecasting (Y. Zhao et al., 2022). Additionally, these econometric methods are also known to be sensitive to changes in data and can potentially result in inaccurate forecasts (Hyndman & Athanasopoulos, 2018a).

2.2.2 Regression Models

Several studies have proposed regression-based models for forecasting workforce needs in varying industries (Bell et al., 2003; Khali Persad & Varghese, 1995). These models estimate the workforce needs based on some independent variables and provide insights into the influence of these variables. Kim et al., (2016) presented the linear regression model for establishing the relationship between workforce requirements and project details such as the cost, type and

degree of urbanization. The study revealed the relationship between the workforce requirements (in terms of person-hours) and the independent variables, with certain capacity improvement projects requiring up to 4.45 times the person-hours compared to pavement projects. The estimated regression model exhibited a goodness of fit of approximately 0.74. Other studies (Bell et al., 2003; Khali Persad & Varghese, 1995; J. M. W. Wong et al., 2011; Yang & Kim, 2019) have presented univariate and multivariate regression methods for estimating the workforce needs in project-level contexts. However, this approach has limitations, particularly when dealing with missing parameters of interest and its inability to capture dynamic systems (Safarishahrbijari, 2018).

2.2.3 Analytical stock-and-flow Models

The stock-and-flow model describes system behavior by incorporating stocks (fluctuating variables) and flows (changes over time) (Forrester, 1958). In the context of workforce forecasting, stocks refer to the workforce level, while flows capture workforce changes such as job promotions, recruitments, and rotations. This modeling approach enables the incorporation of time concepts into the model and offers the advantage of interpretability, making it accessible to policymakers(Safarishahrbijari, 2018).

Sing et al., (2012) developed a labor supply model for the construction industry using the stock-flow approach. The model's performance was evaluated using Hong Kong census statistics and data from 3,000 construction workers. The model was employed to determine future aging distribution trends and workforce supply across different trades. Stock-and-flow models have also found applications in the medical industry, specifically in the long-term projection of crucial health workers like surgeons (Crettenden et al., 2014; Fraher et al., 2013).

2.3 Summary

In addition to the previously discussed models, several other models have also proved to be particularly effective in estimating the workforce in the construction industry. Some of these include the labor multiplier method (Chan et al., 2006), which assumes a direct relationship between the staff-hour and the cost of the project, and the gray model (Benitez et al., 2013), which is a model based on the first order differential equation. Most of the models come with certain limitations. Most proposed models' prediction accuracy has yet to be tested to verify the models' strength (Y. Zhao et al., 2022). Also, the models do not adapt well to dynamic changes, which could be mitigated by machine learning and AI prediction technology (Y. Zhao et al., 2022).

Chapter 3. Data and Methodology

3.1 Data

The data utilized in this study was obtained from ARDOT. The dataset is comprised of historical records encompassing employee information and projects details spanning a period of 10 years, between 2012 and 2021. Two distinct files were acquired: the employee data file containing relevant information about the employees' work hours, and the project details data file documenting records of previous project details.

3.1.1 Employee data

This data contains logged information about the total number of working hours/days for construction engineers and inspectors working on various projects within the STA. The dataset includes a total of 1,070,643 records that were retrieved for all employees over the study period of 10 years. Each record provides relevant details of the employee at the time of data entry, such as job title position, total time spent on activities for the logged data, and project identifier for the specific project on which activity was carried out for the day. Initial exploration revealed some inconsistencies over the duration covered by the data, such as changes to job title descriptions and incorrectly filled data, which were all reconciled.

Furthermore, a closer analysis revealed significantly lower person-hours recorded for projects in the first year of the dataset (2012). With proper follow-up on this discrepancy with the relevant agency, it was discovered that the introduction of a new data recording system during that period could potentially account for this problem. As a precautionary measure to avoid incomplete person-hour records, all projects from the first year were excluded from the analysis. Consequently, approximately 14.9% of projects were omitted, as they were presumed to have less reliable and incomplete person-hour records.

3.1.2 Project details data

While the employee data provides valuable insights into the workforce, it lacks specific information about the corresponding projects. To address this gap, additional data focusing on the projects themselves were collected. This project details data encompassed essential information about the projects over the same 10-year period as the employee data.

A total of 3,910 projects were gathered, representing a diverse range of 23 different project types. Alongside each project, the data included the respective project's cost, which denotes the total awarded sum for its completion and is assumed to reflect the overall execution expenses. Furthermore, the "Let year," signifying the year in which the project was awarded, was also recorded. Given that projects were both awarded and executed over an extended period, it was necessary to adjust the project costs to a common base year to account for inflation. Therefore, the project costs were adjusted based on the "Let year," which represents the year when the project was awarded, and cost estimation was conducted. In this study, a common base year of 2011 was selected as the earliest project in the dataset was awarded during this year.

In addition to cost and "Let year," the project details data encompassed other significant features. These include the project's start and end dates, status (indicating its progress or completion), and descriptions of activities associated with the project. Collectively, these details provide a comprehensive overview of the projects considered in the analysis.

3.2 Data Cleaning

In this section, the raw data obtained from the ARDOT were subjected to further processing and cleaning to facilitate the forecasting at both the project level and agency level. The details regarding these levels of data cleaning are discussed below.

3.2.1 Project level data

The project level forecast aims to estimate the total person-hours required for the completion of a specific project. To achieve this, a regression problem was formulated, with the project details serving as independent variables and the total person-hours as the target variable. To establish the relationship between project attributes and person-hours, it was necessary to merge the "employee data" and "project details data" files. The unique project identifier present in both datasets served as the primary key for this merging process. However, it was observed that a portion of the projects were still ongoing, and hence, the total person-hours required for their completion were not recorded. These ongoing projects accounted for 16.1% of the entire dataset and were subsequently excluded from the project level dataset.

Upon initial exploration of the project data, it was noted that certain projects exhibited unusually low person-hours. Correspondence with the concerned agency revealed that these projects might have involved the use of external consultants, whose hours were not accounted for in the employee data. As a result, extreme values were removed from the dataset under the assumption that such projects included unaccounted consultant hours. After these exclusions, 1,490 projects were retained for project level forecasting.

To develop the regression model for project level forecasting, independent variables, also known as predictor variables, were selected based on existing literature and the available project attributes. Specifically, the project type, year difference, and adjusted award amount were chosen as the independent variables (Bell et al., 2003; Khali Persad & Varghese, 1995). The project type represents the category under which the construction project is classified, indicating similarities in terms of scope, complexity, and required workforce. The adjusted award amount refers to the project contract amount adjusted for inflation, as no information on the actual execution costs

were available. Past studies have demonstrated that the project cost serves as an effective predictor of total person-hours (Bell et al., 2003; Chan et al., 2006; Kim et al., 2016). Additionally, an ordinal variable representing the reference base year was included to account for differences due to technological advancements and changing practices over time, which may influence workforce requirements for different projects.

The target variable for the machine learning problem at the project level is the total person-hours required to complete the project. This includes the cumulative sum of all employee person-hours logged on the project from the start date to the end date, encompassing various job titles except for resident and assistant resident engineers.

$$\boldsymbol{y^T} = [y_1, y_2, \dots, y_m]$$

$$\boldsymbol{X} = \begin{bmatrix} x_{c,1}, x_{t,1}, x_{y,1} \\ x_{c,2}, x_{t,2}, x_{y,2} \\ \vdots \\ x_{c,m}, x_{t,m}, x_{y,m} \end{bmatrix}$$

Where y_m represents the total person-hours for the mth project, and $x_{c,m}, x_{t,m}, x_{y,m}$ represents the cost, type and year difference of the mth project.

3.2.2 Agency level data

The agency level forecasting pertains to the estimation of the future workforce required by the agency. This can be measured over different durations, such as daily, weekly, monthly, quarterly or yearly intervals (P. H. K. Ho, 2010; J. M. W. Wong et al., 2007). The monthly workforce requirement was adopted for this study. This includes all person-hours expended on the projects undertaken by the agency within a given month.

Upon initial exploration of the monthly person-hours data, notable anomalies were observed in the year 2012 and the last month of the final year, 2021. These months were

excluded, to ensure that only months with complete and reliable data were utilized in the forecasting process. Figure 1 illustrates the monthly hour requirements of all construction engineers and inspectors at ARDOT over the years 2013 through 2021. It is important to note that the person-hour requirements encompass all hours worked on both completed and ongoing projects, except the months with incomplete data, such as July 2021.

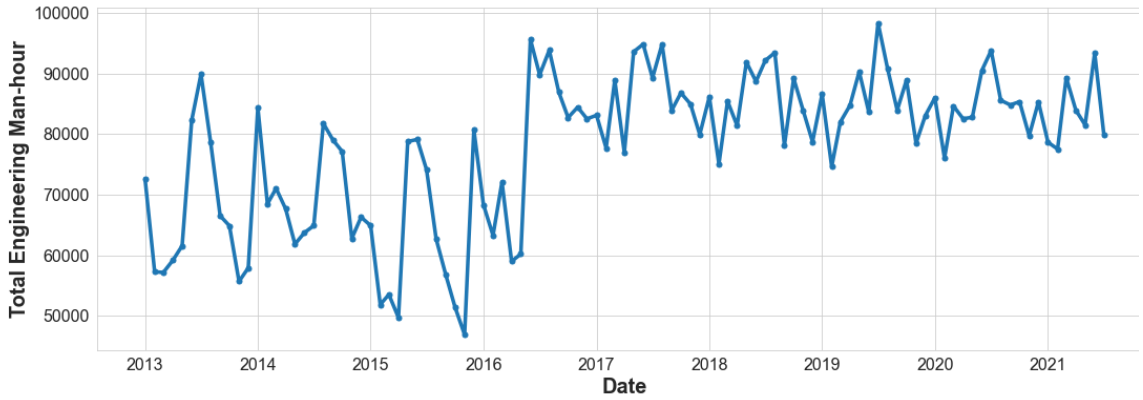

Figure 1: Monthly engineering person-hours between 2013 and 2021

In order to assess the effectiveness of the forecasting models across different time ranges, the data were divided into multiple sets using a train-test split approach. In this context, short-range forecasting refers to the prediction of values that are relatively close in the future, typically less than one year. Conversely, long-range forecasting involves predicting time points significantly farther into the future, extending beyond the one-year threshold. To accommodate both short-range and long-range forecasts within the dataset, separate train-test sets were defined. The short-range set reserved one year of data for testing the model's performance, while the long-range set reserved two and a half years of data for testing the model's effectiveness in longer-term predictions. Consequently, a train-test ratio of 7.5:1 was established for the short-range

forecast, and a ratio of 2.5:1 was employed for the long-range forecast. The breakdown of this division is presented in Table 1.

Table 1: Time series train-test split for short- and long-range forecasting

	Train data	Test data
Short Range	Jan-2013 – July-2020	Aug-2020 – July-2021
Long Range	Jan-2013 – Feb-2019	Mar-2019 – July-2021

3.3 Project level regression models

This section provides a comprehensive description of the statistical and machine learning algorithms explored for the purpose of forecasting the total person-hours required for construction engineers and inspectors in specific projects. The objective was to develop accurate and reliable models that can estimate the workforce needs for individual construction projects.

Various statistical and machine learning techniques were considered in this study to identify the most effective approach for project level forecasting. These algorithms were selected based on their proven applicability in similar domains and their ability to handle regression problems, where the total person-hours serve as the target variable.

The investigated algorithms encompass a range of methodologies, including linear regression, decision trees, random forests, support vector regression (SVR), and neural networks. Each algorithm offers distinct advantages and capabilities, and their suitability for the task at hand was evaluated based on their ability to capture the relationships between project attributes and total person-hours.

By employing these algorithms, the aim was to develop forecasting models that can accurately predict the workforce requirements for construction projects. The insights gained from

these models will assist in resource planning, scheduling, and overall project management, enabling more efficient allocation of human resources and improved project outcomes.

In the subsequent sections, the details of each algorithm, including their underlying principles, implementation strategies, and evaluation techniques are included. By comprehensively investigating these statistical and machine learning approaches, an attempt was made to identify the most effective model for project level forecasting of total person-hours for construction engineers and inspectors.

3.3.1 Linear Regression

Linear regression attempts to model the linear relationship between independent and target variables. While regression models with just one independent variable are referred to as simple linear regression, the ones that employ two or more independent variables are called multiple linear regression models. It is by far the most widely used prediction algorithm and is one of the few techniques that has been adopted in the forecast of person-hour requirements for transportation construction projects (Bell et al., 2003; Khali Persad & Varghese, 1995; Kim et al., 2016). Several variations of the linear regression have been developed to curb the limitations of the classic regression model (Hoerl & Kennard, 1970; Lukman et al., 2019; Swindel, 1976; Tibshirani, 1996). Three variations of these models were employed in addition to the classic ordinary least squared regression model.

3.3.1.1 Ordinary Least Squared Regression

This is one of the previously used methods for predicting employee person-hours for state transportation agencies (Bell et al., 2003; Khali Persad & Varghese, 1995; Kim et al., 2016). It estimates the relationship between independent variables and the target variable (person-hours) by minimizing the sum of the squares of differences between the variables. As a linear model,

one of its basic assumptions is the linear relationship between the independent variables and the target variable. One of the reasons why it is widely used and preferred is its interpretability and prediction accuracy. However, some tasks require less interpretability, and the emphasis lie more on the model's prediction strength, such as in the case of person-hour forecasting. A general representation of the model is shown in equation 3.1. Where y represents the target variable, x_1 through x_p represents the independent variables in the model and β_0 through β_p represents the estimated coefficients for the model.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_p x_p \ldots + \varepsilon \quad \textbf{(3.1)}$$

Studies by (Khali Persad & Varghese, 1995; Kim et al., 2016) have shown a linear relationship between the log transformation of the project cost and the total person-hours. The transformation scales the data values, and hence, the same transformation can be employed for other machine learning models. A representation of the linear model is as shown in Equation 3.2, where $x_{t_1 j}, x_{t_2 j}, \ldots, x_{t_n j}$ represents the indicator variables of project type t_1 through t_n, for all observations j, x_c represents the cost of the project, x_y represents the year difference from the base year, y represents the target variable (total person-hours for given project), β also represents the coefficients of the respective variables.

$$\log(y_j) = \beta_0 + \beta_c \log(x_{cj}) + \beta_{t_1} x_{t_1 j} + \beta_{t_2} x_{t_2 j} + \cdots + \beta_{t_n} x_{t_n j} + \beta_y x_{yj} \ldots + \varepsilon \quad \textbf{(3.2)}$$

$$\boldsymbol{y^T} = [\log(y_1), \log(y_2), \log(y_3), \ldots, \log(y_n)]$$

$$\boldsymbol{X} = \begin{bmatrix} 1, \log(x_{c,1}), x_{t_1,1}, x_{t_2,1}, \ldots, x_{t_n,1}, x_{y,1} \\ 1, \log(x_{c,2}), x_{t_1,2}, x_{t_2,2}, \ldots, x_{t_n,2}, x_{y,2} \\ \vdots \\ 1, \log(x_{c,m}), x_{t_1,m}, x_{t_2,m}, \ldots, x_{t_n,m}, x_{y,m} \end{bmatrix}$$

$$\boldsymbol{\beta^T} = [\beta_0, \beta_c, \beta_{t_1}, \beta_{t_2}, \dots, \beta_{t_n}, \beta_y]$$

$$\boldsymbol{y = X\beta + \varepsilon} \quad \textbf{(3.3)}$$

Given that the estimated parameters are represented by $\widehat{\boldsymbol{\beta}}$ and **e** refers to the residual vector of the model,

$$\boldsymbol{y = X\widehat{\beta} + e}$$

$$\sum \boldsymbol{e_j^2 = e^T e = \left(y - X\widehat{\beta}\right)^T \left(y - X\widehat{\beta}\right)}$$

$$\frac{\partial \sum \boldsymbol{e_j}^2}{\partial \widehat{\boldsymbol{\beta}}} = \boldsymbol{-2X^T y + 2X^T X\widehat{\beta}}$$

$$\boldsymbol{X^T X\widehat{\beta} = X^T y}$$

$$\boldsymbol{\widehat{\beta} = (X^T X)^{-1} X^T y} \quad \textbf{(3.4)}$$

3.3.1.2 Lasso Regression

Lasso regression, another variation of linear regression models, incorporates an additional element called L1 regularization. Unlike ordinary least squares regression, lasso regression includes a penalty term that shrinks the estimated coefficients towards zero. This regularization technique has been widely used to improve the accuracy and reliability of the classic linear regression method and has demonstrated effectiveness in specific problem domains (Tibshirani, 1996). One notable advantage of lasso regression is its performance on small sample sizes, making it well-suited for datasets with limited observations (Tibshirani, 1996; Zou, 2006). In the context of the given problem, where the number of projects is relatively small, lasso regression holds promise. While several advanced techniques and improvements were proposed in recent years (Jarret et al., 2022; Yamada et al., 2014), the classic lasso regression technique was adopted for this study. Estimating the coefficient for lasso regression is shown in Equation 3.5.

$$\widehat{\boldsymbol{\beta}}_{lasso} = \arg\min_{\beta} \left\| \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}} \right\|^2 + \boldsymbol{\lambda}|\widehat{\boldsymbol{\beta}}| \quad \textbf{(3.5)}$$

Where λ is a non-negative parameter for regularization, the second term is the L1 regularization, and other variables remain as previously defined. The lasso regulates the coefficients, allowing the possibility of ~0 value when such a feature is not important to the model.

3.3.1.3 Ridge Regression

The ridge regression is another variation of the classic linear regression model, having an L2 regularization incorporated into it. Like the lasso regression, the regularization term serves as a penalty function to the estimated coefficients, regulating its values. The ridge regression has an advantage regarding its computational cost, as it converges faster when its regularization parameter is utilized (Hoerl & Kennard, 1970). In addition, it has the benefit of handling multicollinearity better, penalizing inefficient coefficients and shrinking their value. Here, this approach might offer speed in terms of training time and better coefficient estimates.

$$\widehat{\boldsymbol{\beta}}_{ridge} = \arg\min_{\beta} \left\| \boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}} \right\|^2 + \boldsymbol{\lambda}|\widehat{\boldsymbol{\beta}}|^2 \quad \textbf{(3.6)}$$

The λ in Equation 3.6 is the non-negative regularization parameter, and the second term, $\boldsymbol{\lambda}|\widehat{\boldsymbol{\beta}}|^2$ is referred to as the L2 regularization.

3.3.1.4 ElasticNet Regression

The elastic net regression combines the power of both the lasso and ridge regression by incorporating the L1 and L2 regularization penalties into the existing linear model. It is mostly famous for its control on overfitting the data and helps reduce variance in the model while maintaining high predictive power (Zou & Hastie, 2005). It assumes a linear relationship

between the target and independent variables and fits the model by minimizing the penalized least squares objective function. Utilizing both regularizations into its model encourages sparsity in its coefficients, as some of its values could be reduced to ~0, hence, simulating feature selection in the model(Zou & Hastie, 2005). In addition, the L2 regularization helps improve the model's stability as it penalizes the model for large coefficients.

$$L(\lambda_1, \lambda_2, \widehat{\boldsymbol{\beta}}) = \|y - X\widehat{\boldsymbol{\beta}}\|^2 + \lambda_1\|\widehat{\boldsymbol{\beta}}\| + \lambda_2\|\widehat{\boldsymbol{\beta}}\|^2$$

$$\widehat{\boldsymbol{\beta}}_{opt} = \arg\min_{\beta}\{L(\lambda_1, \lambda_2, \widehat{\boldsymbol{\beta}})\} \quad \textbf{(3.7)}$$

As seen in Equation 3.7, both L1 and L2 are combined into the linear model. The ability of the model to adapt to nonlinearity is another advantage it offers to the problem set (Wei et al., 2019). It explores possible nonlinear relationships within the independent cost and project type variables with the target variable of person-hours for projects.

3.3.2 Tree Ensemble Models

Tree ensemble models refer to ensemble models that utilize decision trees as their base learners. Ensembles, in general, are a collection of predictive models that are combined to provide a single prediction (Kocev et al., 2013). Past studies have demonstrated the superiority of tree ensemble models amongst machine learning models for prediction (Breiman, 2001; Hastie et al., n.d.; Murphy, 2012). These models leverage the power of decision trees and their ability to capture complex relationships between variables.

The essence of tree ensemble models lies in combining the predictions of multiple decision trees. By aggregating the predictions, the ensemble model aims to reduce variance and improve overall accuracy beyond what each individual tree could achieve. This is achieved through a combination of techniques known as "bagging" and "boosting" (Oza & Russell, 2001).

Bagging, short for bootstrap aggregating, involves the creation of multiple bootstrap training sets by resampling the original training data. Each bootstrap sample is then used to train an individual decision tree, collectively forming the ensemble (Bbeiman, 1996). By generating diverse training sets through resampling, bagging reduces overfitting and enhances the model's generalization ability. Conversely, boosting is more involved and generates a series of base models learned from a weighted training set using decision trees. The models' weights are determined by the error of the preceding model (Freund & Schapire, 1997). Boosting aims to build a strong ensemble by sequentially improving upon the weaknesses of each base learner.

3.3.2.1 Random Forest

The random forest is a classic type of tree ensemble. In this model, diversity in the predictors uses bootstrap replicates similar to bagging and changing the set of descriptive attributes while learning (Breiman, 2001). The bootstrap samples are obtained through the random sampling of the training dataset (with replacement) until there is an equal number of samplings contained in the training set. For each node of the decision tree built using the bootstrap replicates, a random subset of the descriptive attributes is extracted, and the best attributes are selected from the subset (Breiman, 2001). This technique is a powerful one for predicting person-hour requirements as several decision trees are based on the attributes of the project.

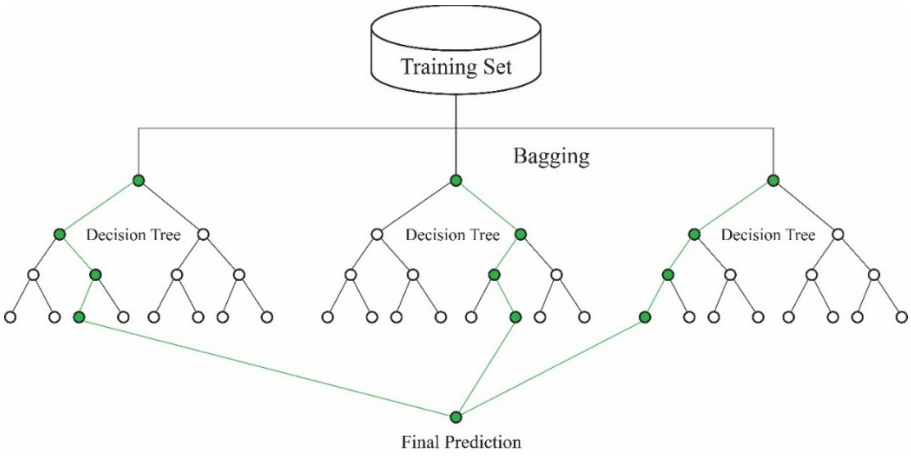

Figure 2: Illustration of the Random Forest Tree Ensemble

A random forest of ensemble *n* trees $\{T_1(\boldsymbol{X}), T_2(\boldsymbol{X}),\ T_3(\boldsymbol{X}) \ldots, T_n(\boldsymbol{X})\}$, where $\boldsymbol{X}$ are vectors of bootstrap sample replicates from the training set. Ensemble output produces n outputs $\{\hat{y}_1 = T_1(\boldsymbol{X}), \ldots\ \hat{y}_n = T_n(\boldsymbol{X})\}$. The final prediction of the random forest model is the average of the predictions $\hat{y} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i$. In the context of forecasting total person-hour requirements for construction engineers and inspectors, the random forest algorithm is deemed valuable due to its ability to capture complex relationships between project attributes and the target variable. By leveraging the collective knowledge of an ensemble of decision trees, the random forest holds promise in delivering accurate and reliable predictions.

3.3.2.2 Gradient Boosting

The gradient boosting technique uses ensemble decision trees and is famous for classification and regression problems. Distinguished from the random forest's utilization of bagging, gradient boosting leverages the boosting technique to construct an ensemble of decision trees. This iterative process involves sequentially incorporating weak learners into the base learner, with each subsequent learner trained to rectify the errors made by its predecessors (Freund & Schapire, 1997; Friedman Jerome H., 2001). By minimizing the loss function with

gradient descent and taking advantage of the collaborative effort of the weak learners, gradient boosting creates an ensemble of models with superior predictive performance. This has proven to be effective, and they offer the additional advantage of being interpretable (Friedman Jerome H., 2001).

Given the problem set as,

$$\boldsymbol{y} = G_1(\boldsymbol{X}) + \boldsymbol{e}_1 \tag{3.8}$$

Where $\boldsymbol{y}$ is the target variable, $\boldsymbol{X}$ is the vector of the independent variables, and $G_1(\boldsymbol{X})$ is a weak learner unable to capture the relationship between $\boldsymbol{X}$ and $\boldsymbol{y}$ fully. This implies that the residual error, $\boldsymbol{e}_1$, will have some correlation with the target variable, $\boldsymbol{y}$.

$$\boldsymbol{e}_1 = \boldsymbol{y} - G_1(\boldsymbol{X})$$

$$\boldsymbol{e}_1 = h_1(\boldsymbol{X}) + \boldsymbol{e}_2$$

Where $h_1(\boldsymbol{X})$ is another weak learner. The ensemble of the two weak learners would yield an estimate of,

$$\hat{\boldsymbol{y}} = G_2(\boldsymbol{X}) = G_1(\boldsymbol{X}) + h_1(\boldsymbol{X}) + \boldsymbol{e}_2 \tag{3.9}$$

Combining several weak learners over *n* iterations:

$$G_n(\boldsymbol{X}) = G_{n-1}(\boldsymbol{X}) + h_{n-1}(\boldsymbol{X}) \tag{3.10}$$

The weak learner $h_{n-1}(\boldsymbol{X})$ can be any model, such as linear regression, neural networks, or decision trees. The tree-based model is the most common learner adopted (Zhang et al., 2019) and used in this study. The model attempts to minimize the residual error given by the loss function $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$, and this function is minimized by the gradient descent technique to estimate the optimal parameters.

The gradient descent helps to find the minimum of a function and is a first-order iterative optimization algorithm (Freund & Schapire, 1997; Ruder, 2016). While the conventional approach for gradient descent is to estimate the gradient with respect to the parameters of the model, the gradient boosting calculates the gradient with respect to the predicted value (Friedman Jerome H., 2001; Zhang et al., 2019).

In the context of this study, the application of gradient boosting holds significant promise in forecasting the total person-hour requirements for construction engineers and inspectors. By harnessing the power of ensemble learning and leveraging the strengths of weak learners, gradient boosting presents a robust and interpretable approach to accurately predict workforce demands for transportation construction projects.

Table 2: Gradient boosting pseudocode

Pseudocode	
Initialize the model	$G_0(\boldsymbol{X}) = \dfrac{\sum \boldsymbol{y}}{n}$
Estimate the gradient	$grad_n = \left\{\dfrac{\partial L[\boldsymbol{y}, G_{n-1}(\boldsymbol{X})]}{\partial G_{n-1}(\boldsymbol{X})}\right\}$
Fit weak learners to residuals	$\gamma_n = \dfrac{h_n(\boldsymbol{X}) \cdot [\boldsymbol{y} - G_{n-1}(\boldsymbol{X})]}{h_n(\boldsymbol{X})^2}$
Update model	$G_n(\boldsymbol{X}) = G_{n-1}(\boldsymbol{X}) + \gamma_n \cdot h_m(\boldsymbol{X})$

3.3.2.3 Catboost regression

The catboost regression is an ordered boosting algorithm designed to solve the overfitting problem of boosting algorithms. In ordered boosting, the ensemble trains a model for estimating the residual error from limited data and makes predictions based on the entire data with the ensemble. Furthermore, random permutation is added to the ordered boosting, which helps it overcome overfitting issues. The training speed of the algorithm is improved via feature

combinations where groups of variables with the same information gain are combined in a group. In addition, the catboost algorithm requires minimal parameter tuning for maximum performance in terms of prediction accuracy (Prokhorenkova et al., 2017). The original model proposed has two modes: plain and ordered. The plain mode implements the regular gradient boosting, while the ordered mode takes advantage of the ordered boosting proposed. The categorical variable for implementing the catboost regression needs not to be encoded as it was developed based on categorical features.

In the context of this study, the categorical variables used in the CatBoost regression do not require explicit encoding. The algorithm incorporates categorical features natively, allowing the input vector to represent these features as a single variable (Prokhorenkova et al., 2017). The encoding of categorical variables is described in Equations 3.11 and 3.12, which facilitate the utilization of CatBoost regression for forecasting total person-hour requirements.

$$\boldsymbol{y^T} = [\log(y_1), \log(y_2), \log(y_3), \dots, \log(y_n] \quad \textbf{(3.11)}$$

$$\boldsymbol{X} = \begin{bmatrix} 1, \log(x_{c,1}), x_{t,1}, x_{y,1} \\ 1, \log(x_{c,2}), x_{t,2}, x_{y,2} \\ \vdots \\ 1, \log(x_{c,m}), x_{t,m}, x_{y,m} \end{bmatrix} \quad \textbf{(3.12)}$$

Where x_t is a categorical encoded feature referring to different project types, and other notations remain as previously defined.

By leveraging the strengths of ordered boosting, random permutation, feature combinations, and automatic handling of categorical features, CatBoost regression offers a specialized and effective approach for predicting workforce demands in transportation construction projects. Its ability to address overfitting, enhance training speed, and handle

categorical variables seamlessly makes it a valuable addition to the ensemble of models employed in this study.

3.3.3 Support Vector Regression

Support Vector Regression (SVR) is a powerful algorithm that builds upon the principles of Support Vector Machines (SVM) to address regression problems. SVMs are renowned for their effectiveness in both linear and nonlinear classification tasks, and SVR extends this capability to predict continuous target variables (Drucker et al., 1996). Its main working principle is to find an optimal hyperplane which best fits the training data while maximizing the margin around the predicted values. The model aims to identify a regression hyperplane that closely matches the training instances, known as support vectors, while simultaneously minimizing the prediction error and maintaining a balance between a narrow margin and tolerance for deviations from the actual values. SVR formulates the problem as a convex optimization task and maximizes the margin with minimum tolerance for deviations from actual values. Regularization parameters are incorporated into the SVR to keep the tolerance under control and allow users control over the parameters to strike a balance between underfitting and overfitting the model (Drucker et al., 1996; Smola et al., 2004). This regularization helps prevent over-reliance on specific training instances and promotes generalization to unseen data.

Furthermore, SVR has the capability of addressing nonlinear relationships by employing the use of kernel functions. The functions transform the original space into higher dimensional space allowing the SVR to capture more intricate patterns and integrate nonlinear boundaries into the model. Some kernel functions include polynomial, sigmoid, or radial basis functions (RBF) (Hofmann et al., 2008; Smola et al., 2004). Its versatility makes it valuable in several domains, including finance, environment, traffic forecasting, economics, and clinical research,

where precise and reliable regression predictions are essential (Alwee et al., 2013; Drucker et al., 1996; Hong et al., 2011; Law & Shawe-Taylor, 2017; Simian et al., 2020). The data modification established for the linear models formulation was maintained for the SVR as it attempts to predict the person-hours required to complete given projects.

Given $\boldsymbol{y} = f(\boldsymbol{X}) = \langle \boldsymbol{w} \cdot \boldsymbol{X} \rangle + b$,

Minimize $\frac{1}{2} \|\boldsymbol{w}\|^2$

Subject to $\begin{cases} \boldsymbol{y} - \langle \boldsymbol{w} \cdot \boldsymbol{X} \rangle - b \leq \varepsilon \\ \langle \boldsymbol{w} \cdot \boldsymbol{X} \rangle + \boldsymbol{b} - \boldsymbol{y} \leq \varepsilon \end{cases}$

Where $\boldsymbol{w}$ is the weight vector and b is the bias term.

3.3.6 Decision Trees

Decision trees are very widely used and one of the most popular machine learning techniques for regression problems. They have also served as the base learner in some of the previously described models. The decision trees partition the feature space based on the values of input features, creating a hierarchical structure that resembles a tree (Breiman et al., 1984). Every internal node of the "tree" represents a decision based on a specific feature, while the end nodes, called "leaf" nodes, are the predicted continuous values. They provide an intuitive and interpretable representation of the decision-making process, making them popular for understanding the relationships between the independent and target variables in regression problems.

Decision trees are suitable for tasks that involve both categorical and numerical features. In addition, they can capture nonlinear relationships and interactions within the variables, making it possible to model complex regression patterns (Murphy, 2012). The trees are also robust to outliers and can work well in missing data, as it simply makes decisions based on the available features (Rokach & Maimon, 2007). While they may be susceptible to overfitting when

the trees are more extensive than is required, techniques such as pruning and restricting the tree size and complexity help to curb this in the machine learning technique (Quinlan, 1986).

Decision tree classifiers work by finding a feature split that reduces the impurities in the child nodes the most. For classification problems, this is defined by the Gini impurity or entropy. However, for regression problems, such as in the case of the prediction of person-hours, the mean squared error is used to define the impurity at each feature split (Sebastian Raschka & Vahid Mirjalili, 2019). This is defined as:

$$I(t) = MSE(t) = \frac{1}{N_t} \sum_{i \in D_t} (\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^2 \quad \textbf{(3.13)}$$

Where $N_t, D_t, \hat{y_t}$ refers to the number of training observations, all training subsets and the predicted value at node t.

3.3.7 Neural Networks

Artificial neural networks (ANN) are some of the most widely used machine learning methods with several real-world applications. They are the heart of deep learning algorithms, which have taken the forefront in several research areas, from recommendation algorithms and self-driving cars to language translations and developing novel drugs for the treatment of diseases (Babu Naik et al., 2022; Tsuji et al., 2021; Uszkoreit, 2017). Basic artificial neural networks are built upon the hypotheses and models of how the human brain works and processes complex information (Sebastian Raschka & Vahid Mirjalili, 2019). They consist of interconnected nodes, which are called neurons and are organized in layers. A typical neural network comprises at least three layers: the input layer, one or more hidden layers, and an output layer. They excel at learning intricate patterns and relationships within data, and their strength lies in their capability to process raw data with minimal manual feature engineering.

There are various approaches to which ANNs are trained where weights are adjusted based on the data passed through the network. In order to introduce nonlinearity into the network, activation functions are used in the networks, giving ANNs the ability to capture and process nonlinear relationships within the data. Beginning from the input layer, the training dataset propagates through the network generating an output in a process called "forward propagation." The network weights are adjusted to minimize a predefined loss function through an optimization algorithm such as stochastic gradient descent in a process known as backpropagation.

Given an input matrix of $A^{(in)}$, $\emptyset$ is the activation function, the hidden layer is estimated as,

$$\boldsymbol{A}^{(h)} = \emptyset(\boldsymbol{A}^{(in)}\boldsymbol{W}^{(h)})$$

Each additional hidden layer serves as an input to the next layer. The final output layer is estimated as follows:

$$\boldsymbol{A}^{(out)} = \emptyset(\boldsymbol{A}^{(h)}\boldsymbol{W}^{(out)})$$

The activation function employed in this study is the reLu activation.

$$reLU = f(x) = max\{0, x\}$$

Depending on the number of hidden layers, neural networks can be "shallow" or "deep." While shallow neural networks have only one hidden layer, deep neural networks have multiple hidden layers and require a larger amount of data (Goodfellow et al., 2016). In this study, different neural network architectures were experimented with, and the best results were obtained using an ensemble of two deep neural networks combined with XGBoost. This finding is consistent with previous research, which has shown that ensemble deep learning models tend to outperform individual deep learning models with tabular dataset (Shwartz-Ziv & Armon, 2022).

The first model is a deep neural network of six hidden layers, each with an activation function of reLu. The number of neurons on the first hidden layer is 32, while the remaining five layers have 64 neurons each. The second model has two hidden layers with 32 neuron and 64-neuron layers. The XGBoost model used to combine the predictions was a simple model with a maximum depth of 3 and learning rate of 0.01. An illustration of the architecture is as shown in Figure 3.

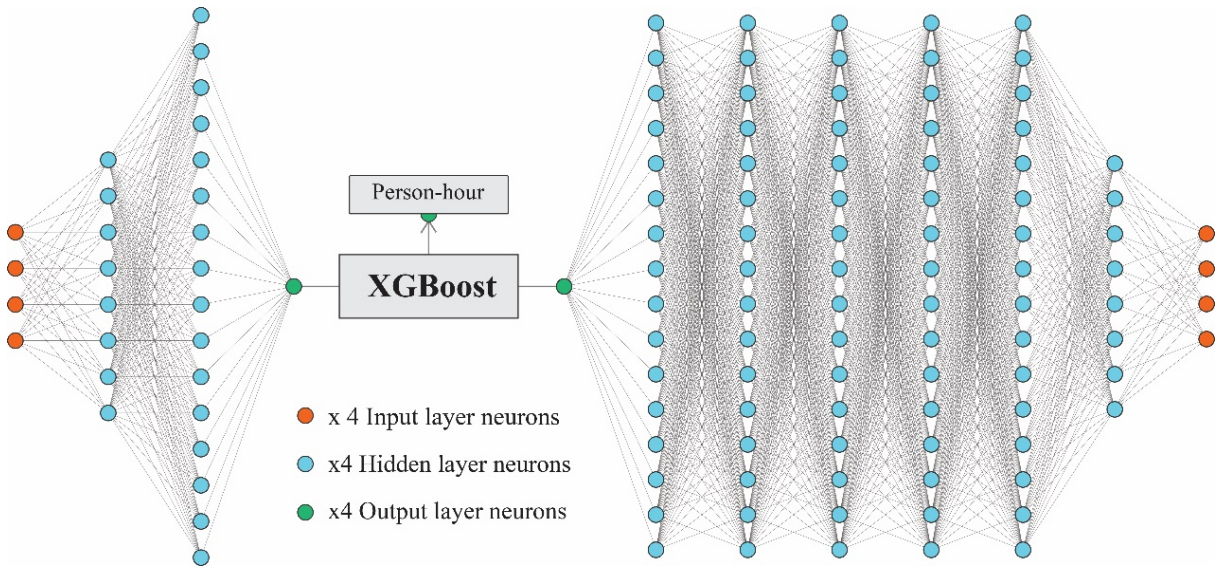

Figure 3: Ensemble Neural Networks layer architecture

3.4 Agency level forecasting models

This section provides a comprehensive description of the Box-Jenkins ARIMA, Long Short-Term Memory (LSTM), and 1-D Convolutional Neural Network (CNN) algorithms, specifically applied to forecast monthly person-hours as a univariate time series. The focus of this analysis is on selecting the most suitable techniques to accurately predict future values based on historical data.

3.4.1 Autoregressive Integrated Moving Average (ARIMA)

The ARIMA model has been the default model for solving time series forecasting problems for several decades (S. L. Ho et al., 2002). It is a statistical model and a generalization of autoregressive (AR) and moving average (MA) models. The AR part of the model forecasts the future based on historical values, while the MA uses the errors from previous predictions. The application of the model typically involves the Box-Jenkins (Box & Jenkins, 1976) method, which includes identification of the order of the model, estimation of the parameters of the model, diagnostic checking of the model and finally, forecasting of future values (Hyndman & Athanasopoulos, 2018b). The order of the model can be denoted as ARIMA (p, d, q) where p, d, q, refers to the orders of the autoregressive, differencing and moving average polynomials respectively. When accounting for seasonality, the order becomes denoted as ARIMA $(p, d, q)(P, D, Q)$ where P, D, Q represents the seasonalcomponent of the orders.

One of the conditions for using the ARIMA model is the need for stationarity in the time series data. This could be inspected visually and tested using the Augmented Dickey-Fuller (ADF) test (Dickey & Fuller, 1979) to determine if the time series data is indeed stationary. Adfuller test carried out on the data and its visual observation reveals that it is nonstationary, at a p-value of 0.727. Therefore, the data had to be transformed to achieve stationarity. Differencing is a commonly used technique to address nonstationarity in time series data, involving the subtraction of successive time periods in the series. Multiple differencing operations can be performed until stationarity is achieved. In our analysis, first differencing was sufficient to achieve stationarity for the monthly person-hour data. Figure 4 displays a plot of the differenced series, clearly showing a constant mean across the series. The ADF test conducted on the differenced series confirmed stationarity with a p-value of approximately 0.

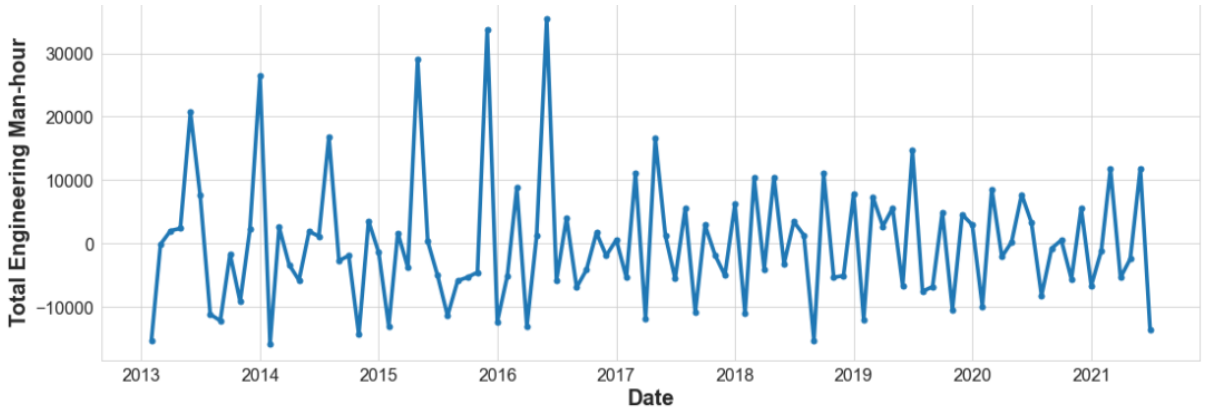

Figure 4: First differencing of monthly person-hours between 2013 and 2021

Determining the orders of the model could be done in a variety of ways, and a stepwise search using the Akaike information criterion (AIC) was used to select the model for the problem at hand. The most appropriate model was established to be one of order ARIMA(2,1,2)(1,0,1)$_{12}$. The paramters for the model were estimated using the training data and are included in the appendix. The seasonal decomposition of the model can also be seen in Figure 5.

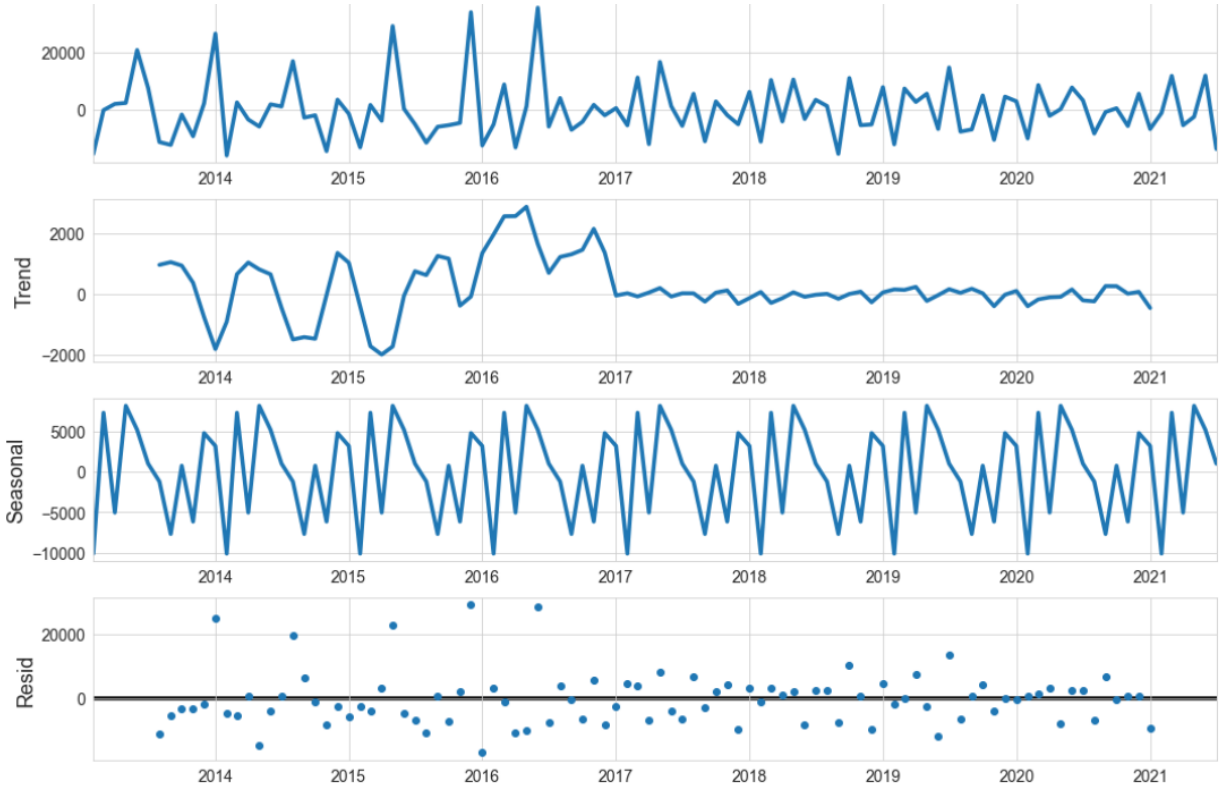

Figure 5: Decomposition of the time series

3.4.2 Long Short-Term Memory

Long short-term memory (LSTM) neural networks are a type of recurrent neural network (RNN) that is widely used in the field of deep learning (Cho et al., 2014; Hochreiter & Schmidhuber, 1997). They have been specially designed to address the limitations of traditional RNNs by capturing long-term dependencies in sequential data. This is achieved through memory cells and gating mechanisms that enable the network to retain and selectively update information over a long period. The critical feature of LSTMs is their ability to learn when to forget or remember specific information, making them supper effective for time series prediction. The architecture of the LSTM network consists of interconnected memory cells, each with an input, forget and output gate. The gates control the flow of information and allow the flow of information while filtering out noise and irrelevant inputs. The input gate decides the part of the information to be added while the forget gate selectively discards information (Gers et al., 2000). The forget gate, f_t, input gate, i_t, and output gate, o_t, are as estimated in the equations below (Sebastian Raschka & Vahid Mirjalili, 2019).

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{xf}\boldsymbol{x}^{(t)} + \boldsymbol{W}_{hf}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{xi}\boldsymbol{x}^{(t)} + W_{hi}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i\right)$$

$$\boldsymbol{o}_i = \sigma\left(\boldsymbol{W}_{xo}x^{(t)} + \boldsymbol{W}_{ho}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o\right)$$

The combination of the gates and memory cells helps the LSTM overcome the vanishing (exploding) gradient problem known to plague the traditional RNNs (Hochreiter & Schmidhuber, 1997). Other methods of solving this problem include gradient clipping and truncated backpropagation through time (TBPTT); however, the LSTM has proved superior to these other methods (Sebastian Raschka & Vahid Mirjalili, 2019). LSTM has demonstrated astonishing performance in a wide range of real-world applications, making them an excellent tool for

modeling complex sequential data like time series data (Chen et al., 2015; Urgen Schmidhuber et al., n.d.; Yunpeng et al., 2017). They excel at learning and utilizing seasonality information over time, which is advantageous for forecasting tasks. Moreover, LSTMs are robust enough to handle noisy data, a common characteristic of time series data.

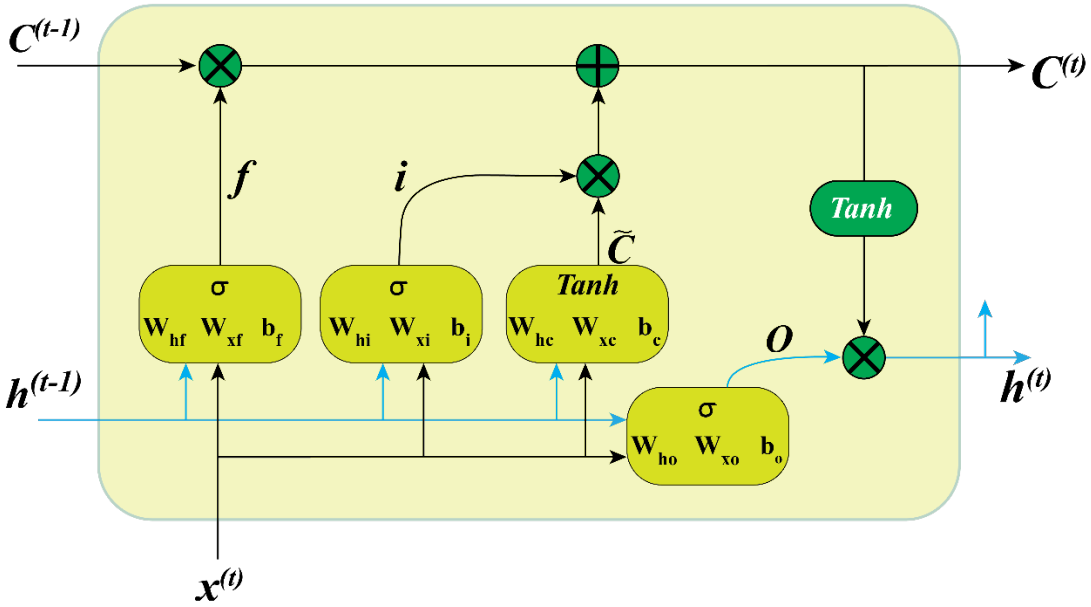

Figure 6: LSTM neuron

Several parameters of the LSTM architecture were experimented with, and the best-performing network was observed to have four hidden layers. A single layer with 32 LSTM neurons yielded optimal results, as increasing the number of LSTM neurons only increased network size and required more iterations to converge. The remaining three hidden layers were fully connected layers with regular neurons, each activated with the rectified linear unit (ReLU) function, and comprised 64, 32, and 16 neurons, respectively. Figure 7 provides an illustration of the LSTM network architecture. Notably, this network consistently produced the best fit for the time series data, irrespective of the forecast range.

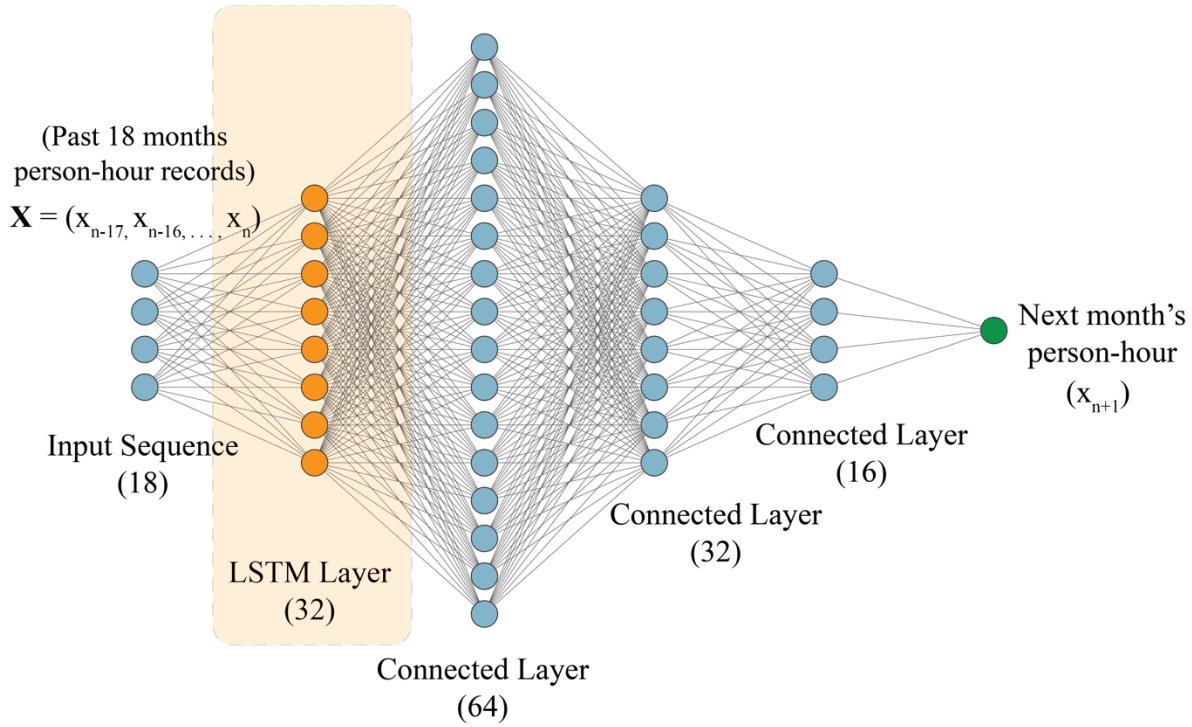

Figure 7: LSTM Architecture for time series forecasting

3.4.3 One-dimension Convolutional Neural Networks (1-D CNN)

Convolutional Neural Networks (CNN) are renowned for their image-processing capabilities. Initially inspired by the operation of the cortex of the human brain in recognizing objects, they have been widely successful in computer vision (Fukushima, 1980; Hubel & Wiesel, 1959; Sebastian Raschka & Vahid Mirjalili, 2019). Its one-dimensional form, however, has gained significant attention in recent years due to its effectiveness in processing sequential data. Some of its applications are in audio signals (Mustaqeem & Kwon, 2019), natural language processing (Xiang & Song, 2020) and time series (B. Zhao et al., 2017) related tasks. A typical CNN model consists of convolutional and related layers, as well as fully connected MLP (Multilayer Perceptron) layers. The convolutional layers extract essential features from the raw input data, which are then fed into the MLP layers for classification or regression..

The complete neural network architecture starts with the input layer. This layer contains $N \times k$ neurons, where k refers to the number of features and N, the input length fed into the network at each instance. k assumes the value of 1 for a univariate time series, leaving the input layer with a total N neurons. The input layer is directly connected to the convolutional layer. This layer performs convolution operations with filters of predefined size, l. Other parameters required for the convolution include the stride, s, which refers to the number of units through which the filter slides over the layers, and the padding, d, which helps to control the size of the output layer from the convolution operation. The output neurons after each convolution can be estimated by equation 3.14. The output of the convolution operation is passed to the pooling layer, which down samples the convolutional layer. The pooling layer reduces the extracted feature's dimensionality and helps maintain only the most essential details in the network. This is usually achieved by estimating the average of the neurons or selecting the maximum value within sets of neurons. Typical CNN architectures are made of several convolutional and pooling layers. The final output from these layers is consolidated into a feature layer that the fully connected MLP can easily process.

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - l}{s} \right\rfloor + 1 \qquad \textbf{(3.14)}$$

The MLP layers are regular neural networks receiving their input from the feature layer of the convolutional network. Like several deep learning architectures, it can have several hidden layers with varying activation functions before finally being connected to the output layer that gives the final prediction.

Several experimentations were carried out with the 1-D CNN, and the best-performing network is shown in Figure 8. As illustrated, two sets of convolutional layers were included with pooling layers before the result is fed into the fully connected layers of neural networks. The

convolutional layers were made up of 1-dimension 128 neurons and 64 neurons, respectively. The fully connected layers following this was a single layer of 100 neurons with an activation function of reLu which was connected to a linear output of a single neuron.

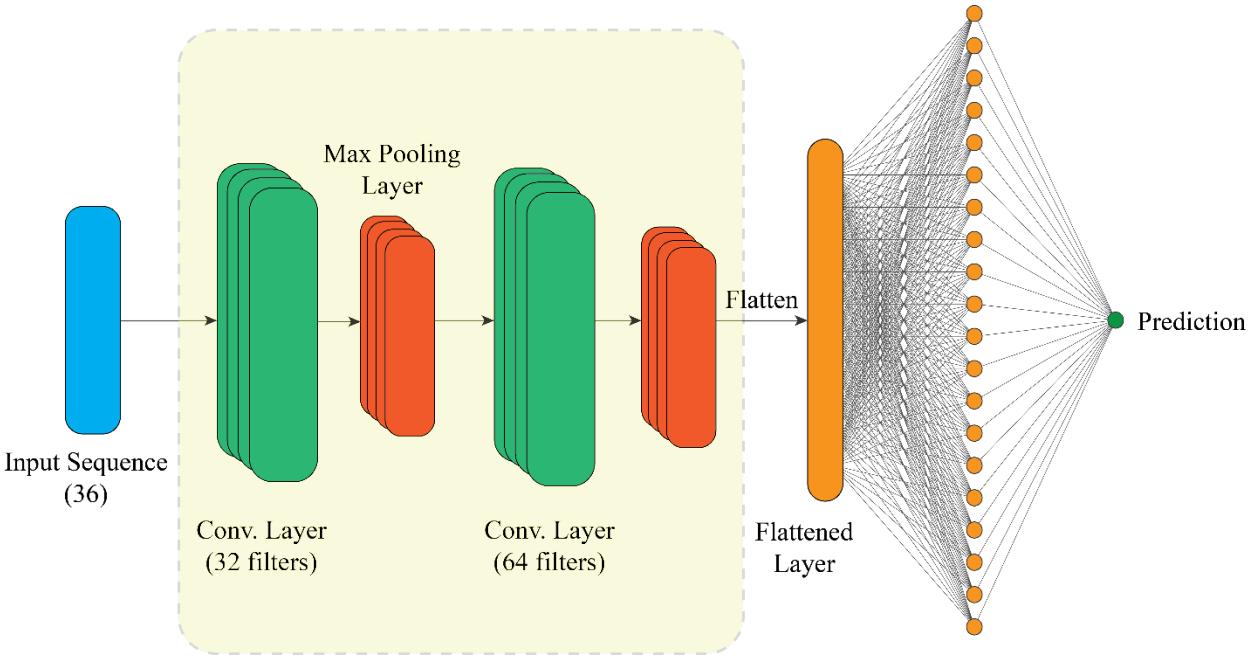

Figure 8: 1-D CNN model architecture for time series forecasting

3.5 K-Fold cross-validation

This technique was used for the project level model as the agency level data are sequential. The machine learning regression models employ the K-fold technique which ensures the model generalizes and does not just perform well on just one test set. The K-fold technique works by portioning the data into K folds, training the model on "K minus one" folds of the entire dataset and testing the remaining single fold. This is repeated K times with a different set used as the test set at each instance. The final performance of the model is the average score across the K folds. The choice of K usually depends on the dataset size and computational resources available. Popular values of 5 or 10 are usually used, and a value of 10 folds was

employed for this study, given the dataset size. A choice value of 10 also ensures enough output samples for hypothesis tests.

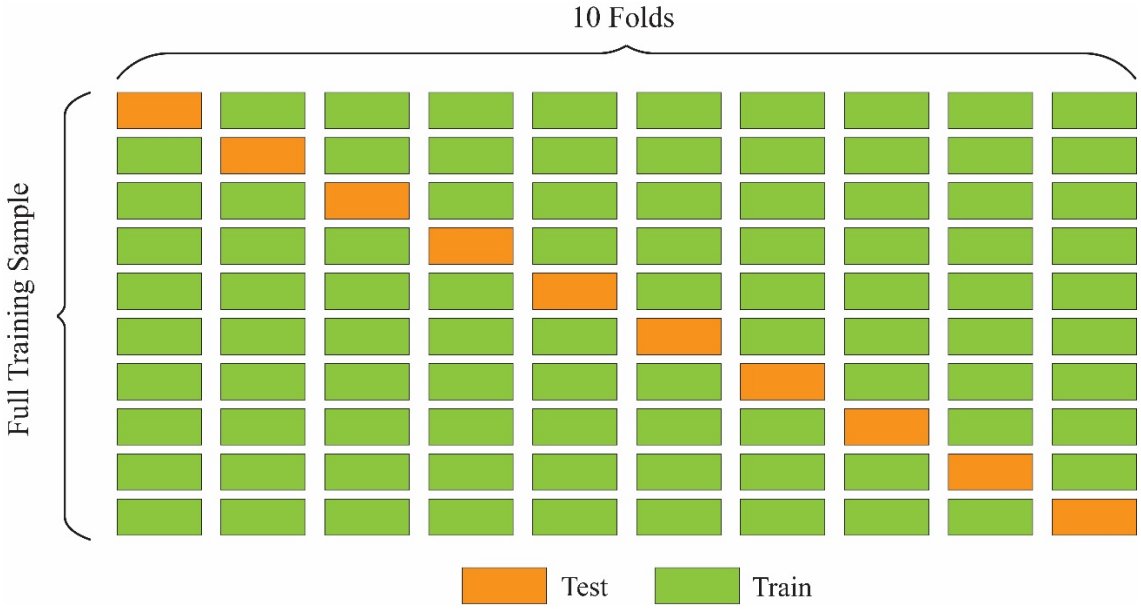

Figure 9: 10-Fold cross-validation train/test set

3.6 Hyperparameters Tuning

Hyperparameter tuning plays an essential role in evaluating the machine learning models utilized. It involves searching for optimal values for the hyperparameters used in each model. Unlike model parameters, which are learned from the data during training, hyperparameters are predefined settings that affect the model's behavior. Common examples of these parameters are the learning rate, regularization parameters, number of trees in tree-based models and many more. Several techniques are used for tuning parameters ranging from the manual tuning of the parameters based on previous literature to automated techniques via search algorithms. The Bayesian optimization, which is efficient, especially when the search space is large, and complex is one of the methods used for tuning. Other optimization techniques for selecting hyperparameters, such as grid search and random search methods were also considered where the

Bayesian optimization might not be favorable. These methods explore the parameter space in a brute-force approach which can be limited in its searches. The best parameters for the models were considered regardless of the tuning approach and the search space and best tuning methods for each model is as detailed in the appendix.

Additionally, to ensure unbiased evaluation during testing using the K-Fold method, a random selection process was employed to determine the project used as the validation set for tuning project level models, while training set of the agency data was further split into train-validate set using 20% of the later dates for validation. Careful selection of the validation set helped minimize bias during model testing and ensures a fair assessment of their performance.

3.7 Accuracy Metrics

This study considers five metrics for evaluating the accuracy of the models compared. These metrics include the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Median Absolute Percentage Error (MAPE), Max Error (ME) and goodness of fit (R^2).

3.7.1 Root Mean Squared Error (RMSE)

This is one of the widely used metrics for regression models. It is estimated as the square root of the average squared difference between actual target values and estimated values. RMSE assigns more weight to more significant errors making it sensitive to outliers in evaluating regression models. It is estimated by taking the square root of the sum of the square of the residuals of each estimation divided by the total number of observations.

$$RMSE = \sqrt{\frac{1}{N}\sum(\boldsymbol{y} - \widehat{\boldsymbol{y}})^2}$$

3.7.2 Mean Absolute Error (MAE)

The mean absolute error estimates the average absolute difference between the actual target values and the estimated value from regression models. Unlike the root mean squared error, this metric is less sensitive to outliers. It is obtained by summing the absolute residuals and dividing the resulting sum by the number of observations.

$$MAE = \frac{1}{N}\sum|\boldsymbol{y} - \widehat{\boldsymbol{y}}|$$

3.7.3 Median Absolute Percentage Error (MAPE)

This is predominantly used for forecasting models. It measures the median percentage difference between the forecasted value and the actual value with respect to the actual value of such measurement. This form of metrics is resistant to outliers and provides a percentage-based value, making it relatively easier to read and interpret. It is calculated by taking the median of absolute percentage errors.

$$MAPE = Median\left\{\left|\frac{\hat{y} - y}{y}\right|\right\}$$

3.7.4 Goodness of fit (R^2)

The model's goodness of fit is measured by the coefficient of determination (R^2), and it assesses the proportion of the variance in the dependent variable, which is explained by the independent variables in a regression model. The possible values for the coefficient of determination range between 0 and 1, where 0 indicates a model poorly explains the variance, and 1 shows a model that is a perfect fit. R^2 is estimated by comparing the sum of squares of residuals to the total sum of squares.

$$R^2 = 1 - \frac{SSR}{SST}$$

$$SSR = \sum (\boldsymbol{y} - \hat{\boldsymbol{y}})^2$$

$$SST = \sum (\boldsymbol{y} - \bar{\boldsymbol{y}})^2$$

3.8 Hypothesis Testing

In this study, hypothesis testing was conducted to compare the performance of machine learning algorithms. Non-parametric tests were used to test the project-level machine learning models as demonstrated in past studies (García et al., 2010; Khadse et al., 2020; Luengo et al., 2009; Trawinski et al., 2012). Although a high number of samples could validate the use of parametric tests due to central limit theory, the samples for project-level models with a k-fold of 10 prediction samples would not meet such requirements. Friedmann's test was therefore conducted on the output to test the significance difference across the models used. The agency-level forecasting has a significant number of outputs to satisfy the central limit theory; hence, parametric testing was considered for testing its results.

3.9 GPU implementation

The total run time of the model training and prediction depends on the computation machine's performance. All models were trained using the same computer with a consistent configuration. Training of deep learning models took advantage of parallel computing by enabling libraries such as PyTorch. NVIDIA GeForce GTX 1660 SUPER with 6.0GB dedicated GPU memory and 16.0GB shared GPU memory. The computer processor is an Intel Core i7-10700K running at 3.80GHz with 32.0 GB memory.

Chapter 4. Results and Discussion

4.1 Project-level forecasts

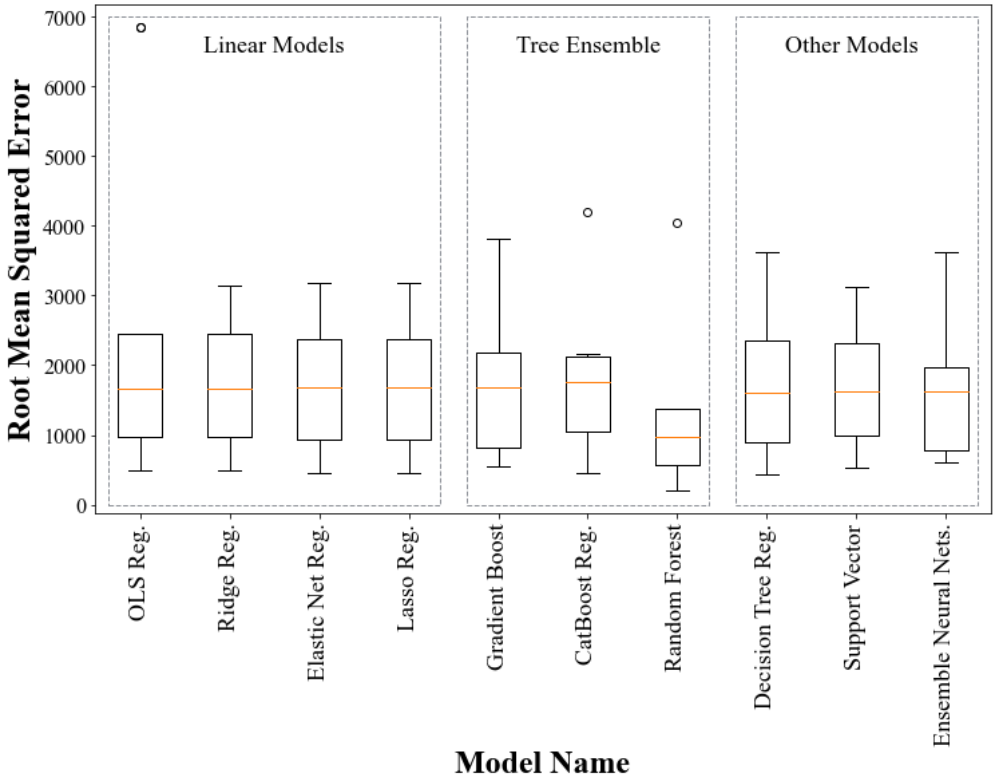

Figure 10: Mean Squared Error of machine learning algorithms

4.1.1 Accuracy

Figures 10 to 14 show the boxplot depicting the performance metrics of the various models in term of prediction accuracy. Each metric is unique and focuses on various strengths of the model. Friedman tests show statistical significance differences in the performance of all models at the 95% significance level. This indicates that at least one of the models performed significantly better or worse than some of the other models.

The linear models' performance are comparable, with each model showing similar variance and accuracy across the metrics. The ordinary least squared regression shows some outlier errors, which was not observed in the other modelsand can be attributed to the presence of

the regularization parameters in the models. Further investigation into these error predictions indicate its occurrence across models with low representation in the dataset.

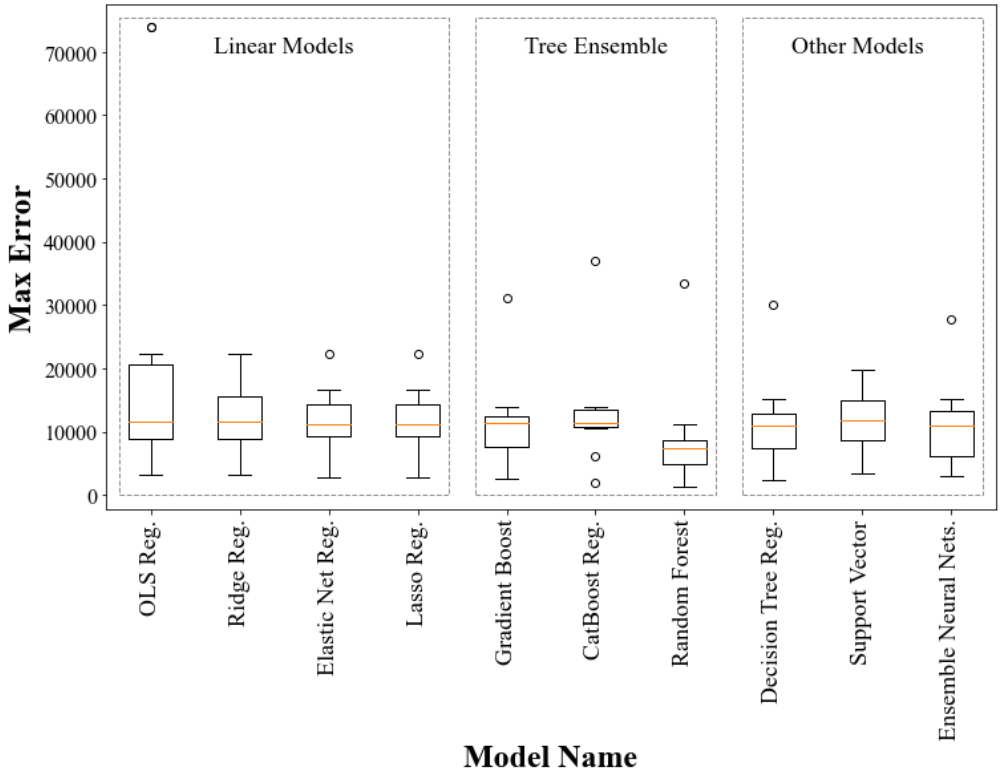

Figure 11: Maximum Error of Machine Learning Algorithms

The tree ensemble models performed best with the random forest model, exceeding every other model across all accuracy metrics. In addition, the variance in its accuracy is relatively small compared to other models, showing stability in its performance. The R-squared value achieved by the random forest model (0.91) exceeds all other models and those established with regression methods in previous studies (Khali Persad & Varghese, 1995).

Single deep learning models and shallow neural networks performed poorly on this tabular dataset. However, the ensemble model of multiple deep neural networks proved effective, as presented by (Shwartz-Ziv & Armon, 2022). Attempts to increase accuracy by adding more deep-learning models to the ensemble yielded only a marginal result, leaving the ensemble

model with two neural networks as the best performing ensemble. Across the accuracy metrics, it can also be seen to achieve the second-best result with a mean R squared of 0.80.

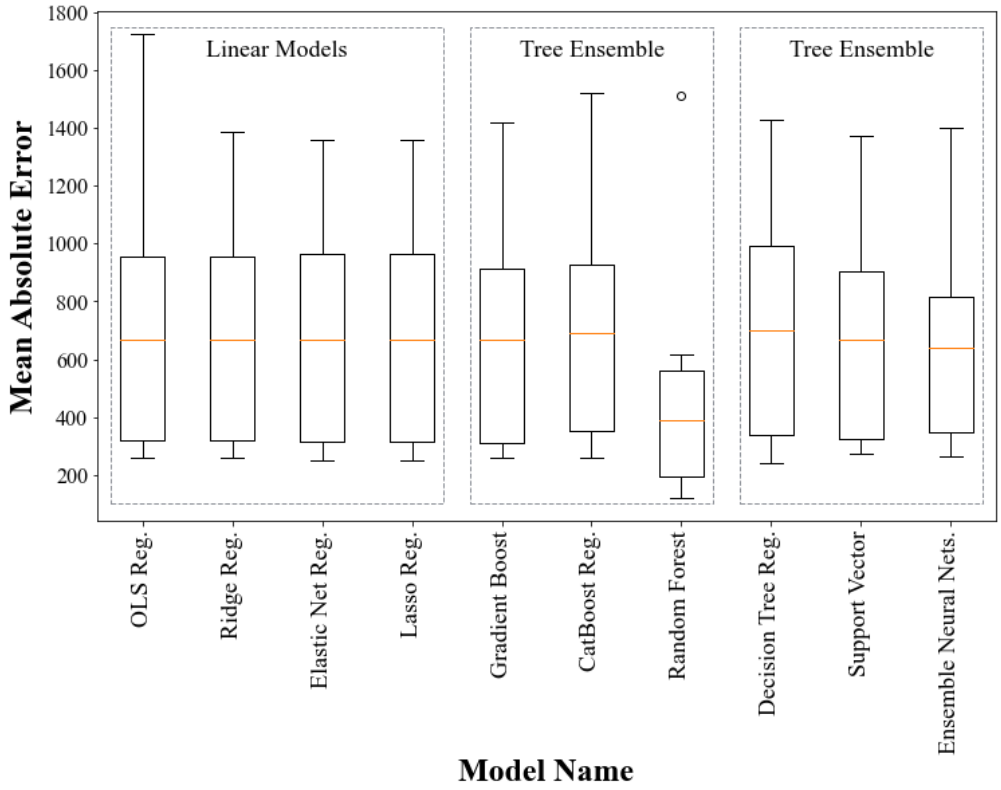

Figure 12: Mean Absolute Error Across Machine Learning Models

Regarding hyperparameter tuning, a combination of automated tuning methods, such as grid search and Bayesian optimization, was used. Interestingly, some parameters optimized through grid search outperformed those optimized using Bayesian optimization, contrary to previous studies. The best-performing parameters were selected regardless of the tuning method. For more detailed information on the hyperparameters and parameter search space, you can refer to the Appendix, which provides additional information on the range of parameters used for tuning.

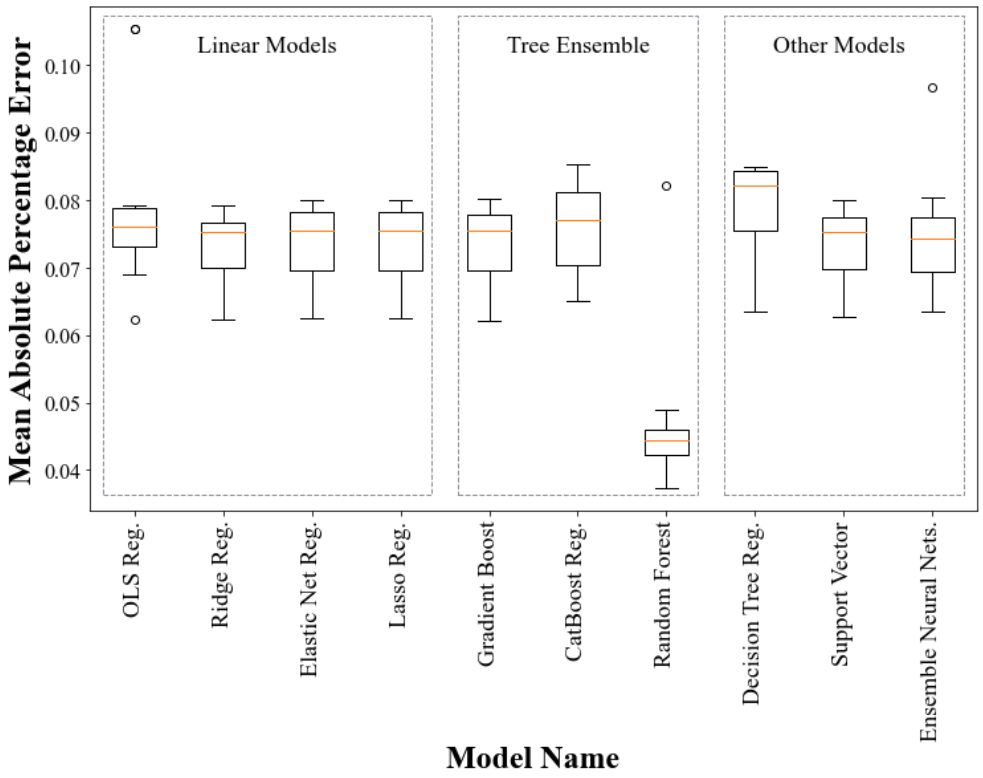

Figure 13: Mean Absolute Percentage Error Across Machine Learning Models

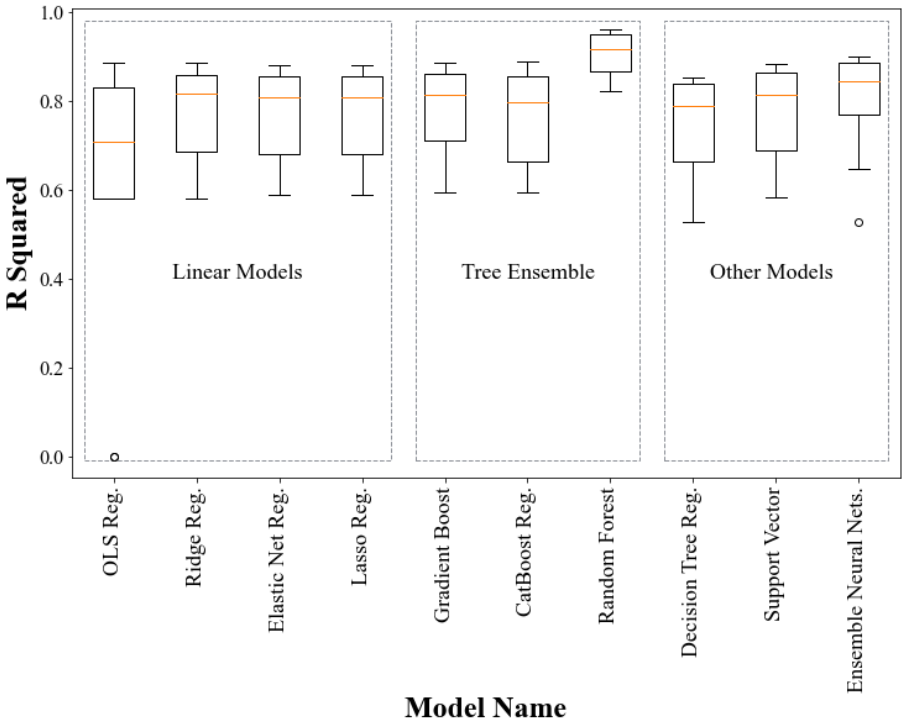

Figure 14: Goodness of Fit across Machine Learning Models

4.1.2 Model training time

Results for the modeling time of the models are shown in Figure 15. As expected, the time taken for the ensemble neural network models to model the data is significantly higher than that for the other models. This is due to the processes involved in training a neural network, having to go through multiple iterations of forward and backward propagation over the entire dataset. The other models are comparable in terms of the time required for training the models. On average, the linear models can be seen to perform significantly better than the other models. This might be due to the simple technique involved in their mode of operation, especially when compared to other models like tree ensembles which relies on training multiple decision trees in its algorithm.

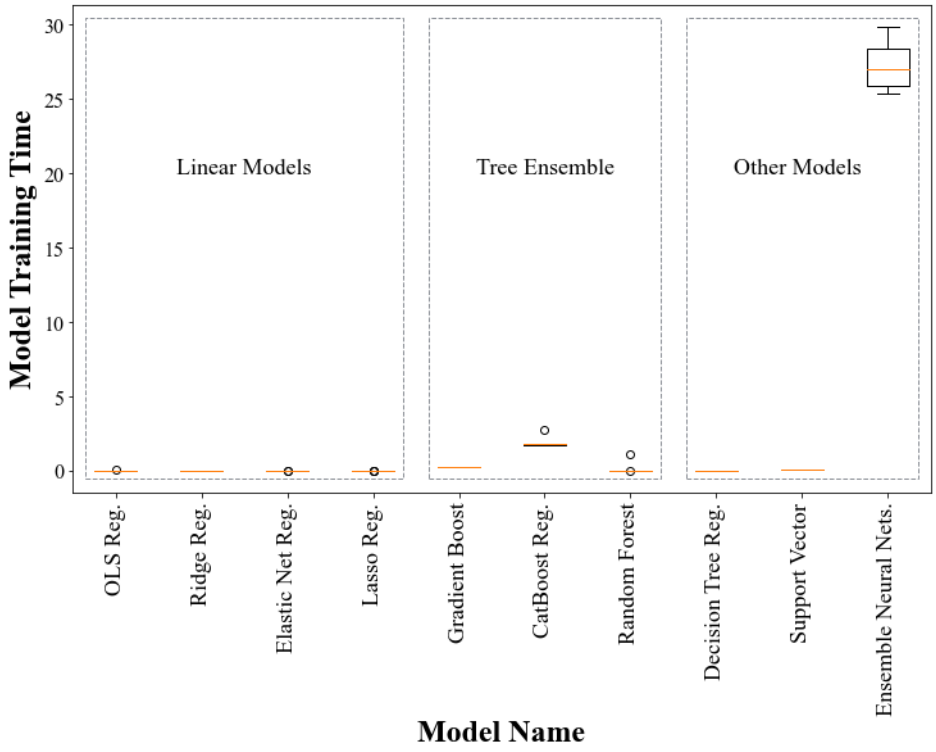

Figure 15: Training Time Across All Models

4.1.3 Prediction Time

The prediction time for the ensemble neural networks takes significantly longer time for its output. While some other machine learning models are ensembles themselves, the neural network takes significantly higher time due to the size of the models involved. On average, the next model with a high prediction time is the random forest model, which is also an ensemble model. This could explain why it takes more time to generate a prediction, as all the ensemble trees need to generate their predictions first. However, with a prediction time of about 0.012 sec for predicting 10% of the total dataset projects, its superior accuracy made it more desirable regardless of its comparatively slow prediction time.

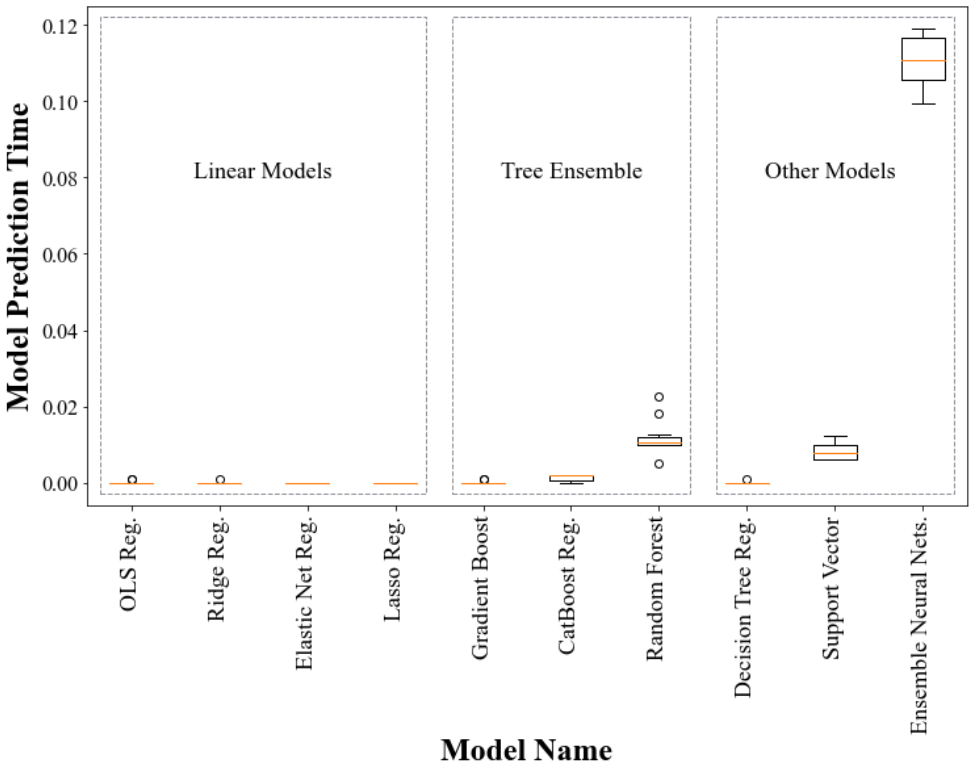

Figure 16: Prediction Time Across All Models

4.2 Agency level

In the agency level forecast analysis, multiple runs of neural network-based models were conducted to generate forecasts. These runs produced varying results due to randomizations present in the models' operations, even when given the same input. To ensure statistical significance in comparing the models, fifty model runs were performed for each model, providing enough samples for parametric test comparisons. While non-parametric tests are suitable for comparing machine learning models, parametric tests would be sufficient when there are enough samples due to the central limit theory (Trawinski et al., 2012). Each model was tested based on their forecasting accuracy over a short-range period of 1 year and a longer-range period of 2.5 years. Two sample t-tests show significant differences between the model output of the LSTM and 1-D CNN at the 99% confidence level.

4.2.1 ARIMA

The ARIMA model yields a consistent result, hence, does not require multiple runs for its predictions. The trend in the short-range forecasting was impressive and captured the variability through the 1-year period. Its average MAE was 3,885 person-hours, which, compared to the average monthly person-hour requirements, yields about 4.98% error. Also, regarding the RMSE metric, its error is about 6.45%, which indicates that the predictions are consistent with little outlier predictions, which would have given rise to a much higher RMSE score. However, compared to its long-range prediction, the ARIMA does not perform as much. It fails to capture the variability of the actual forecasts beyond a certain threshold, as seen in Figure 22. This might be due to the limitations of the traditional models, as the longer-range forecast involved making predictions based on fewer data points with the train-test ratio for the long-range forecast being 2.5 to 1. As the forecast timeline extended, the model tended to produce constant predictions

over time. The MAE for the error in long-range forecasting was larger compared to the short-range forecast, reflecting the greater extent of the forecast and the challenges faced by the ARIMA model in capturing long-term variability.

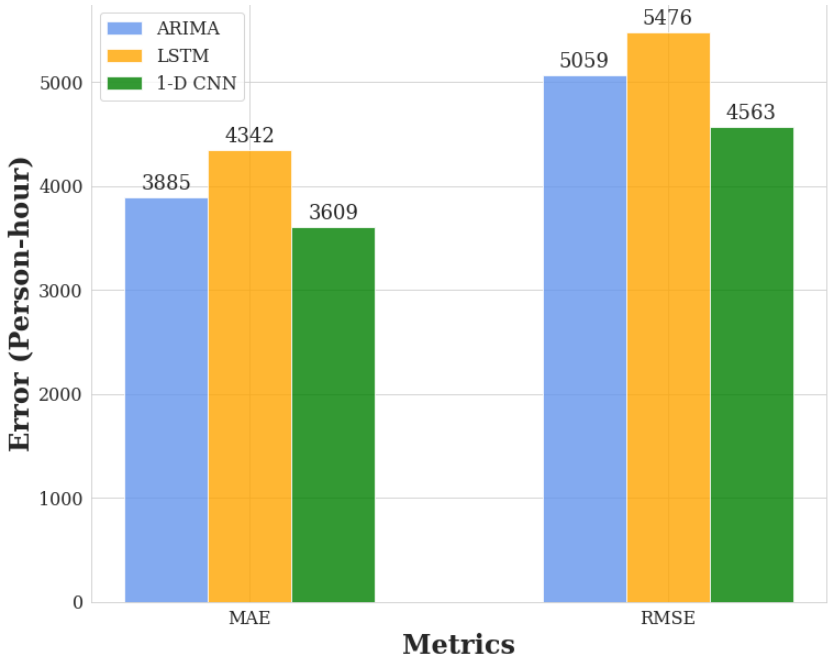

Figure 17: Short-range average accuracy metrics

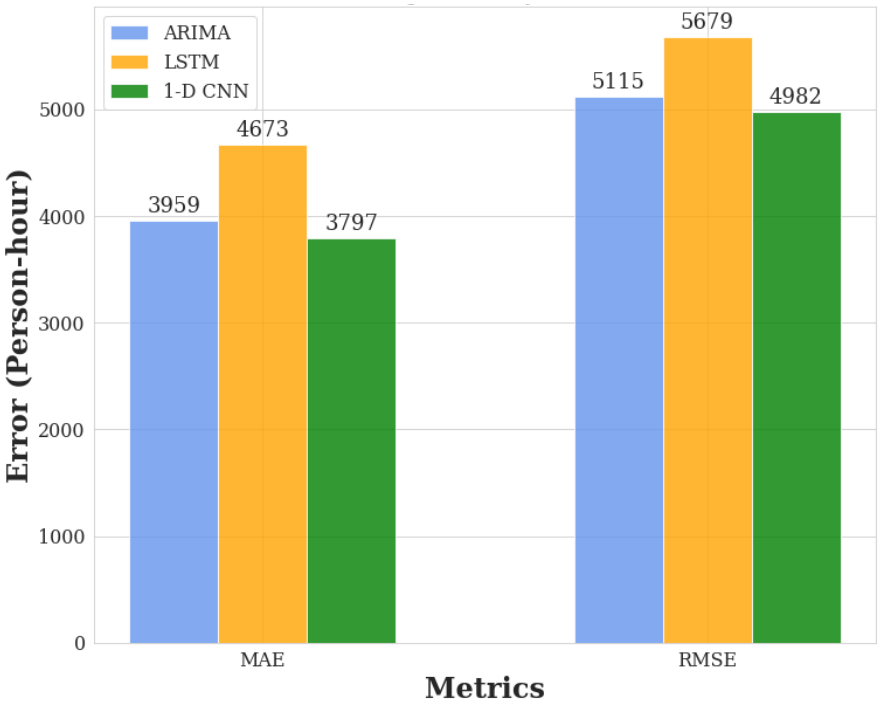

Figure 18: Long-range average accuracy metrics

4.2.2 LSTM

The LSTM model result predictions resulted in the highest error in terms of accuracy. The prediction graph shows significant variability as it attempts to forecast varying results for each time point. The high variance depicted in the distribution of the forecasting error of the model (Figure 19 & 20) indicates it is less stable for forecasting. The average MAE shows 5.57%, while the RMSE gives an error of 7.03% over the short-range forecasting error.

Similar to the other models, the accuracy metrics for the short-range forecasting were better when compared to the longer-range forecasting. Also, regarding the variability of the results, a similar variance was recorded across both forecasting ranges. This might be a result of the best-performing model architecture for both ranges being similar, with the short-range model having an additional hidden layer in its fully connected layer architecture.

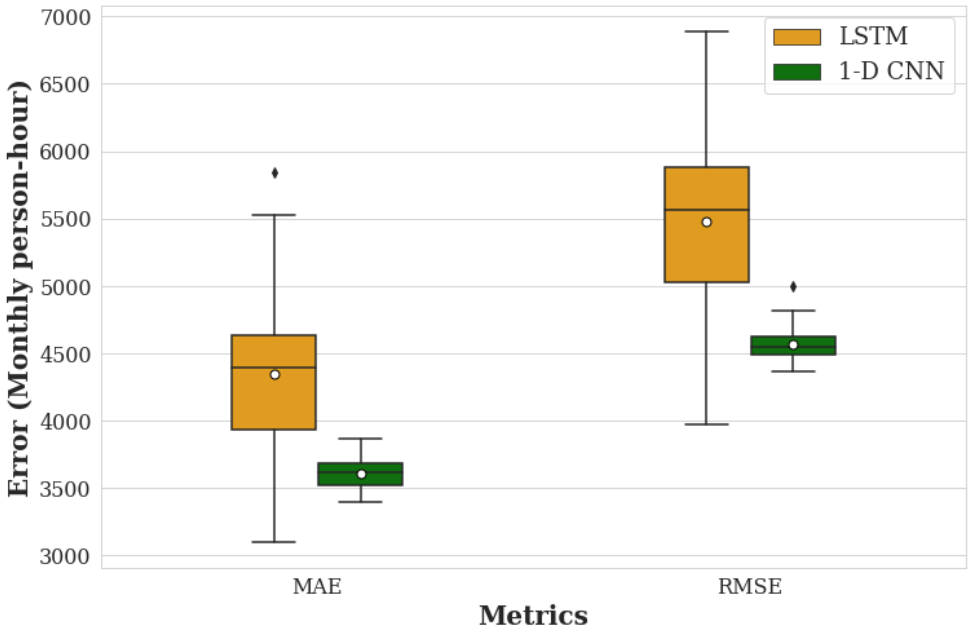

Figure 19: Box plot for short-range forecasting error

4.2.3 1-D CNN

The 1-D CNN model demonstrated the best performance among the models, with minimal variance in its predictions as shown in Figures 20 and 21. Short- and long-range

forecasting accuracy measures were less than 5% for the MAE metrics, with values of 4.60% and 4.87%, respectively. Regarding the RMSE metrics, its forecasting error was 5.85% and 6.39% over short-range and long-range, respectively.

The low standard variation of the model's accuracy, at 124.41 person-hours, indicates that the predictions were consistently close to the actual values. The coefficient of variation, calculated as the ratio of the standard deviation to the mean, was 0.0016, indicating a very low level of variability in the model's accuracy.

Compared to the LSTM model, the 1-D CNN model had a different architecture and input configuration. The 1-D CNN model utilized double the input for prediction in its optimized model, which may have contributed to its better stability. Additionally, the ability of CNN models to detect patterns and serve as feature engineers likely played a significant role in its success in accurately forecasting the monthly person-hour requirements.

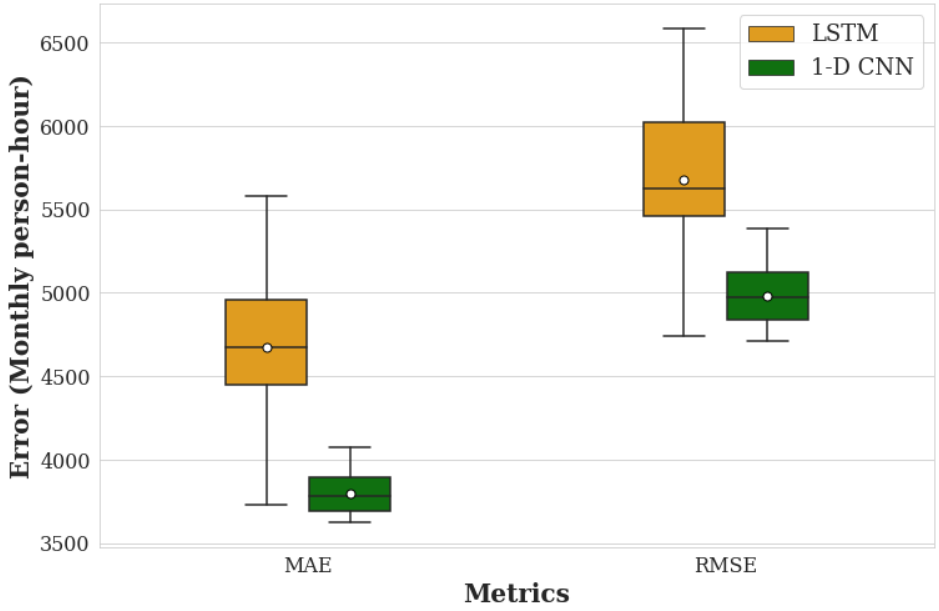

Figure 20: Boxplot showing long-range forecasting error

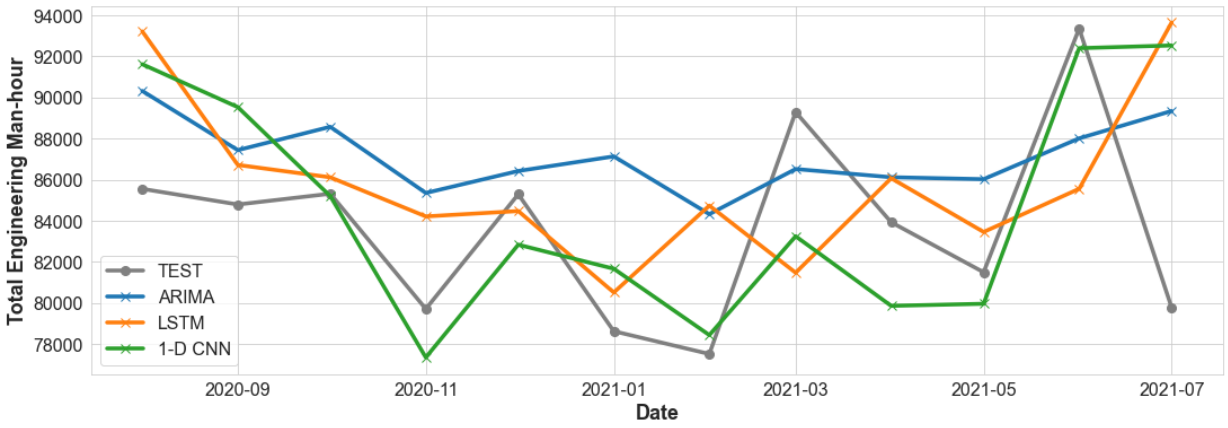

Figure 21: Short-range forecasting error comparison

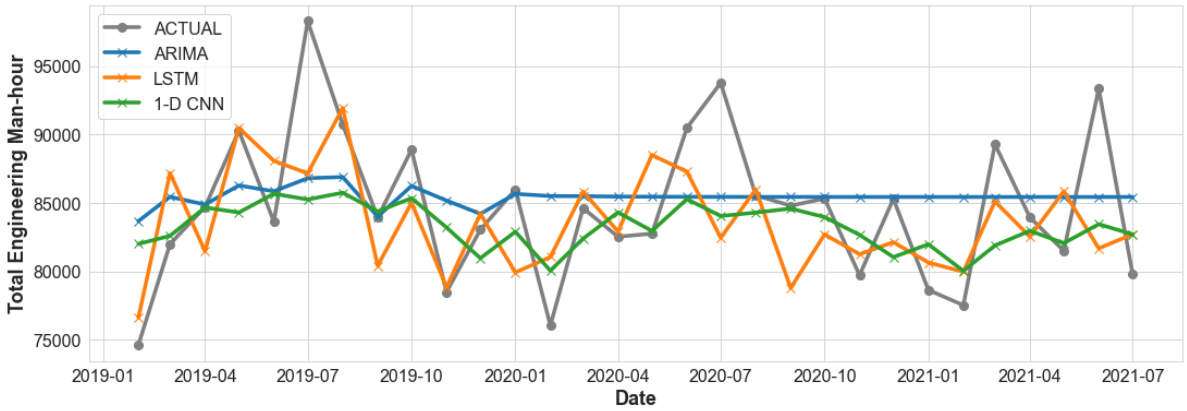

Figure 22: Long-range forecasting error comparison

Chapter 5. Conclusion

Summary

This thesis compared the performance of machine learning models and classic statistical methods in forecasting the level of workforce at the project and agency level. The goal of the thesis is to provide stronger forecasting alternatives for engineers and planners at STAs for their planned projects. To fulfill this objective, data from ARDOT, a STA case study containing information about the employee person-hour details and project details were utilized in building forecasting models for the project-level and agency-level forecasting.

5.1 Project-level

This study developed selected machine learning models to compare their performance across various metrics to forecast person-hour requirements in transportation projects at the project level. Ten Models were developed, which include the ordinary least square regression, Lasso regression, Ridge regression, Random Forest regressor, Support Vector Regressor, Gradient Boost Regressor, Catboot Regressor, Decision Tree regressor and an ensemble deep neural network. Each model uses project attributes such as the project type, cost, and year to predict the total person-hours that would complete such project. Ten-fold cross-validation was conducted to assess the performance of the model's accuracy, training time and prediction time.

The random forest regressor, a tree ensemble model, outperforms the other models with an average MAPE of 4.75% and R squared value of 0.91 due to its superior modeling capability of learning important features to forecast the person-hour requirements of transportation projects. It can also be seen to be stable as there is less variance across the accuracy of the predictions generated by the model. This could be due to the attribute of the random forest model to average its variance based on the results from multiple decision tree. The training time required to model

the data is comparable to the other models with an average time of less than one tenth of a second. The prediction time, when compared to the other model, is quite higher except for the ensemble neural networks; however, this is of less importance as it takes less than one-hundredth of a second to make its prediction on 10% of the project data.

All linear models all gave similar results regarding their accuracy performance, modeling time and prediction time. However, the ordinary least squared regression performed worse compared to the other linear models particularly on underrepresented project types. The other linear models seem to address this issue with the presence of regularizers, which controls the coefficients for such projects. Methods used in previous studies using linear regression involve training different linear regression models for different types of projects (Bell et al., 2003; Kim et al., 2016), which would not take the advantage of shared properties across the project types and less accurate models for project types with less historical projects.

The ensemble neural networks model performed reasonably well, having the second-best average goodness of fit of 0.80, after the random forest model. While deep neural networks are known to have subpar performance with tabular data, the ensemble neural networks, as proposed in (Shwartz-Ziv & Armon, 2022), proved to be applicable to predicting the person-hour requirements for STAs. The downside of this, however, is the long training time which is significantly higher than that of the other model. While its prediction time takes the most time, it is still under two-tenths of a second. Possible improvement might be obtained from the model if given more training data, as the dataset used for this study might be insufficient for such deep neural networks model.

The results could serve as a guidance for STAs in other cities, states, and municipalities for building forecasting tools at the project level. It could help guide the selection of appropriate

models and as a starting basis for building software tools for the forecast of the amount of person-hours required for projects. This would yield an improved workforce planning across the agencies in allocating resources.

5.2 Agency level

Several time series approaches, such as the vector error correction and the box Jenkins ARIMA have been previously used to forecast construction engineer person-hours on the national and agency level workforce forecasting. Very little has been attempted to implement a more complex approach to the time series modelling of the workforce in the construction industry. This thesis presents a univariate time series forecasting of the monthly person-hours requirements of construction engineers and inspectors in a STA over short- and long- range time periods. Neural network based models were compared to the classic box Jenkins ARIMA approach in terms of accuracy.

The ARDOT case study data was transformed into a monthly person-hour time series spanning across almost ten years. Test data was reserved for both short- and long- range forecasting strength purpose and the models developed using the training set. Due to the randomness introduced by the neural networks based models, multiple runs of the optimized models were trained for a more accurate representation of the models.

Amongst the neural network-based models, the 1-D CNN performed better across all metrics and forecasting ranges with a RMSE of 5.85% and 6.39% across the short- and long-range tests. The variance noted across its forecasting error was relatively small compared to the LSTM model, making it more desirable for such forecasting. The LSTM did not perform as expected and this might be as a result of the limited dataset limiting its use of its memory. Also, while the LSTM produced a worse forecasting error than the ARIMA, it can be seen that the

model continues to capture the variability of monthly forecasts over the long- range unlike the ARIMA model. The CNN architecture performs better in extracting essential features from the data, which might have been an edge in its forecasting capability. Similar CNN models could be adopted by STAs across cities, states and municipalities for a more accurate workforce forecast. This provides valuable insight for human resource departments and planners to quickly prepare for future workforce needs adequately with a more reliable forecast.

5.3 Limitations and Future Research

While this study demonstrated the strength of machine learning models in forecasting STA construction engineer and inspector needs, it comes with some limitations. The data size used for each model is limited to the 1,490 projects collected from ARDOT. Some of the models compared would perform significantly better when more data are used for its training. An example of such a model is the neural network ensemble model for the project-level and the LSTM model for the agency level. Also, this study considered only data from one STA, and might be insufficient to generalize the results from this study across all STAs. However, the demonstrated accuracy of the models show their potential across the industry, and more research into its application across several STAs could validate the findings in this research.

Furthermore, details about the project were limited, as more attributes, such as the project's complexity, would provide more features for the machine learning models to learn from. This could be a new feature defined by STAs for describing projects depending on how complex the operations are, regardless of the project type. It should also be noted that the modeling time compared does not consider the level of effort put into tuning the model's hyperparameters which affects the performance of each model. While some models are relatively easier to tune and take less time, others require combining multiple techniques to tune their

parameters and obtain the best-performing optimized parameters. Future research could also adopt interpretable machine learning methods to better understand how the project attributes affect the level of workforce required for its completion. Insights gained here would be useful to planners and engineers in the timely execution of their projects.

The agency level forecast shows a significant improvement with the 1-D CNN, but other models that use the combined power of the 1-D CNN and LSTM have not been considered. Studies (Moskolai et al., 2020; Wahid et al., 2022; Zaheer et al., 2023) have shown the accuracy of such models and this could serve as a great starting point for future research. While CNN helps to extract and maintain the essential features of the network, the LSTM helps to retain long-term information relevant to forecasting. Additionally, the current study focused on univariate time series models for the agency level forecast, which consider only a single variable (person-hour requirements) for forecasting. Future research could extend the analysis to include multivariate time series models. By incorporating additional relevant variables, such as project characteristics, economic indicators, or weather data, the forecasting models can capture the influence of multiple factors on person-hour requirements. This can result in more comprehensive and accurate predictions, enabling better decision-making in workforce planning and resource allocation.

Acknowledgment

The author thanks the Arkansas Department of Transportation (ARDOT) for providing the relevant data used in this study and partially funding this project. The work was performed in part under contract TRC 2101 with ARDOT.

The contents of this thesis reflect the views of the author who is responsible for the facts and the accuracy of the data presented herein. This thesis does not constitute a standard, specification, or regulation.

References

Alwee, R., Hj Shamsuddin, S. M., & Sallehuddin, R. (2013). Hybrid Support Vector Regression and Autoregressive Integrated Moving Average Models Improved by Particle Swarm Optimization for Property Crime Rates Forecasting with Economic Indicators. *The Scientific World Journal*, *2013*, 951475. https://doi.org/10.1155/2013/951475

Azimi Nayebi, B., Mohebbifar, R., Azimian, J., & Rafiei, S. (2019). Estimating nursing staff requirement in an emergency department of a general training hospital: Application of Workload Indicators of Staffing Need (WISN). *International Journal of Healthcare Management*, *12*(1), 54–59.

Babu Naik, G., Ameta, P., Baba Shayeer, N., Rakesh, B., & Kavya Dravida, S. (2022). *Convolutional Neural Network Based on Self-Driving Autonomous Vehicle (CNN)* (pp. 929–943). https://doi.org/10.1007/978-981-16-7167-8_68

Bbeiman, L. (1996). *Bagging Predictors* (Vol. 24).

Bell, L. C., Asce, F., & Brandenburg, S. G. (2003). *Forecasting Construction Staffing for Transportation Agencies*. https://doi.org/10.1061/ASCE0742-597X200319:3116

Benitez, R. B. C., Paredes, R. B. C., Lodewijks, G., & Nabais, J. L. (2013). Damp trend Grey Model forecasting method for airline industry. *Expert Systems with Applications*, *40*(12), 4915–4921.

Box, G. E. P., & Jenkins, G. M. (1976). Time series analysis: Forecasting and control San Francisco. *Calif: Holden-Day*.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time Series Analysis*. Wiley. https://doi.org/10.1002/9781118619193

Breiman, L. (2001). *Random Forests* (Vol. 45).

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*.

Bryant, D. R., Maggard, M. J., & Taylor, R. P. (1973). Manpower planning models and techniques: A descriptive survey. *Business Horizons*, *16*(2), 69–78. https://ideas.repec.org/a/eee/bushor/v16y1973i2p69-78.html

Chan, A. P. C., Chiang, Y. H., Mak, S. W. K., Choy, L. H. T., & Wong, J. M. W. (2006). Forecasting the demand for construction skills in Hong Kong. *Construction Innovation*, *6*(1), 3–19. https://doi.org/10.1191/1471417506ci106oa

Chen, K., Zhou, Y., & Dai, F. (2015). *A LSTM-based method for stock returns prediction: A case study of China stock market*. 2823–2824. https://doi.org/10.1109/BigData.2015.7364089

Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 103–111. https://doi.org/10.3115/v1/W14-4012

Choudhury, J. P., Sarkar, B., & Mukherjee, S. K. (2002). Forecasting of engineering manpower through fuzzy associative memory neural network with ARIMA: a comparative study. In *Neurocomputing* (Vol. 47). www.elsevier.com/locate/neucom

Chung, S. H., Jung, D. C., Yoon, S. N., & Lee, D. H. (2010). A dynamic forecasting model for nursing manpower requirements in the medical service industry. *Service Business*, *4*(3–4), 225–236. https://doi.org/10.1007/s11628-009-0092-4

Collings, D. G., & Mellahi, K. (2009). Strategic talent management: A review and research agenda. *Human Resource Management Review*, *19*(4), 304–313.

Crettenden, I. F., McCarty, M. V, Fenech, B. J., Heywood, T., Taitz, M. C., & Tudman, S. (2014). How evidence-based workforce planning in Australia is informing policy development in the retention and distribution of the health workforce. *Human Resources for Health*, *12*(1), 7. https://doi.org/10.1186/1478-4491-12-7

Dalkey, N., & Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. *Management Science*, *9*(3), 458–467.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*(366a), 427–431.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support Vector Regression Machines. In M. C. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in Neural Information Processing Systems* (Vol. 9). MIT Press. https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf

Edwards, J. S. (1983). A Survey of Manpower Planning Models and Their Application. *Journal of the Operational Research Society*, *34*(11), 1031–1040. https://doi.org/10.1057/jors.1983.234

Forrester, J. W. (1958). Industrial dynamics: a major breakthrough for decision makers. *Harvard Business Review*, *36*(4), 37–66.

Fraher, E. P., Knapton, A., Sheldon, G. F., Meyer, A., & Ricketts, T. C. (2013). Projecting Surgeon Supply Using a Dynamic Model. *Annals of Surgery*, *257*(5), 867–872. https://doi.org/10.1097/SLA.0b013e31826fccfa

Freund, Y., & Schapire, R. E. (1997). *Journal of Computer and System Sciences s SS1504 journal of computer and system sciences* (Vol. 55).

Friedman Jerome H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, *180*(10), 2044–2064. https://doi.org/10.1016/j.ins.2009.12.010

Garza, A., Kumara, S., Martin, J. D., & Finke, D. A. (2013). Multi-agent system for manpower modeling. *IIE Annual Conference. Proceedings*, 187.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gordon, T. J., & Helmer, O. (1964). *Report on a long-range forecasting study*.

Hagopian, A., Mohanty, M. K., Das, A., & House, P. J. (2012). Applying WHO's "workforce indicators of staffing need" (WISN) method to calculate the health worker requirements for India's maternal and child health service guarantees in Orissa State. *Health Policy and Planning*, *27*(1), 11–18. https://doi.org/10.1093/heapol/czr007

Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). *Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction*.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3). https://doi.org/10.1214/009053607000000677

Hong, W.-C., Dong, Y., Zheng, F., & Lai, C.-Y. (2011). Forecasting urban traffic flow by SVR with continuous ACO. *Applied Mathematical Modelling*, *35*(3), 1282–1291. https://doi.org/10.1016/j.apm.2010.09.005

Ho, P. H. K. (2010). Forecasting Construction Manpower Demand by Gray Model. *Journal of Construction Engineering and Management*, *136*(12). https://doi.org/10.1061/ASCECO.1943-7862.0000238

Ho, S. L., Xie, M., & Goh, T. N. (2002). *A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction*. www.elsevier.com/locate/dsw

Hsu, C.-C., & Sandford, B. (2007). The Delphi Technique: Making Sense Of Consensus. *Practical Assessment, Research and Evaluation*, *12*.

Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, *148*(3), 574.

Hyndman, R. J., & Athanasopoulos, G. (2018a). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., & Athanasopoulos, G. (2018b). *Forecasting: principles and practice*. OTexts.

Jarret, A., Fageot, J., & Simeoni, M. (2022). A Fast and Scalable Polyatomic Frank-Wolfe Algorithm for the LASSO. *IEEE Signal Processing Letters*, *29*, 637–641. https://doi.org/10.1109/LSP.2022.3149377

Khadse, V. M., Mahalle, P. N., & Shinde, G. R. (2020). Statistical Study of Machine Learning Algorithms Using Parametric and Non-Parametric Tests. *International Journal of Ambient Computing and Intelligence*, *11*(3), 80–105. https://doi.org/10.4018/IJACI.2020070105

Khali Persad, B. R., & Varghese, K. (1995). *PEER-REVIEWED PAPER FORECASTNG ENGINEERING MANPOWER REQUIREMENTS FOR HIGHWAY PRECONSTRUCTION ACTIVITIES*.

Kim, D. Y., Persad, K. R., Khwaja, N. A., & Chi, S. (2016). Assessment of staffing needs for construction inspection. *KSCE Journal of Civil Engineering*, *20*(7), 2598–2603. https://doi.org/10.1007/s12205-016-0817-6

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, *46*(3), 817–833. https://doi.org/10.1016/j.patcog.2012.09.023

Kwak, N. K., McCarthy, K. J., & Parker, G. E. (1997). A human resource planning model for hospital/medical technologists: An analytic hierarchy process approach. *Journal of Medical Systems*, *21*, 173–187.

Lagarde, M., & Blaauw, D. (2009). A review of the application and contribution of discrete choice experiments to inform human resources policy interventions. *Human Resources for Health*, *7*(1), 62. https://doi.org/10.1186/1478-4491-7-62

Laslett R. E. (1972). A Survey of Mathematical Methods of Estimating the Supply of and Demand for Manpower. *Engineering Industry Training Board, U.K*.

Law, T., & Shawe-Taylor, J. (2017). Practical Bayesian support vector regression for financial time series prediction and market condition change detection. *Quantitative Finance*, *17*(9), 1403–1416. https://doi.org/10.1080/14697688.2016.1267868

Linstone, H., & Turoff, M. (1975). The Delphi Method: Techniques and Applications. In *Technometrics* (Vol. 18). https://doi.org/10.2307/3150755

Lomas, J., Stoddart, G. L., & Barer, M. L. (1985). Supply projections as planning: A critical review of forecasting net physician requirements in Canada. *Social Science & Medicine*, *20*(4), 411–424. https://doi.org/10.1016/0277-9536(85)90020-6

Luengo, J., García, S., & Herrera, F. (2009). A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests. *Expert Systems with Applications*, *36*(4), 7798–7808. https://doi.org/10.1016/j.eswa.2008.11.041

Lukman, A. F., Ayinde, K., Siok Kun, S., & Adewuyi, E. T. (2019). A Modified New Two-Parameter Estimator in a Linear Regression Model. *Modelling and Simulation in Engineering*, *2019*, 6342702. https://doi.org/10.1155/2019/6342702

Martin, V., Hurn, S., & Harris, D. (2012). *Econometric Modelling with Time Series*. Cambridge University Press. https://doi.org/10.1017/CBO9781139043205

Moskolai, W., Abdou, W., Dipanda, A., & Kolyang, D. T. (2020). Application of LSTM architectures for next frame forecasting in Sentinel-1 images time series. *ArXiv Preprint ArXiv:2009.00841*.

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective*.

Mustaqeem, & Kwon, S. (2019). A CNN-assisted enhanced audio signal processing for speech emotion recognition. *Sensors*, *20*(1), 183.

Oza, N. C., & Russell, S. J. (2001). Online Bagging and Boosting. In T. S. Richardson & T. S. Jaakkola (Eds.), *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics* (Vol. R3, pp. 229–236). PMLR. https://proceedings.mlr.press/r3/oza01a.html

Parker, B., & Caine, D. (1996). Holonic modelling: human resource planning and the two faces of Janus. In *International Journal of Manpower* (Vol. 17).

Park, S.-H., Lee, S. M., No Yoon, S., & Yeon, S.-J. (2008). A dynamic manpower forecasting model for the information security industry. *Industrial Management & Data Systems*, *108*(3), 368–384.

Pate, J. E. (1943). Mobilizing manpower. *Social Forces*, *22*(2).

Prescott P. A. (1991). Forecasting requirements for health care personnel. *Nursing Economics*, *9*(1), 18–24.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). *CatBoost: unbiased boosting with categorical features*. http://arxiv.org/abs/1706.09516

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/BF00116251

Rokach, L., & Maimon, O. (2007). Decision Trees. In *Data Mining and Knowledge Discovery Handbook* (pp. 165–192). Springer-Verlag. https://doi.org/10.1007/0-387-25465-X_9

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. http://arxiv.org/abs/1609.04747

Safarishahrbijari, A. (2018). Workforce forecasting models: A systematic review. *Journal of Forecasting*, *37*(7), 739–753. https://doi.org/10.1002/for.2541

Sebastian Raschka, & Vahid Mirjalili. (2019). *Python Machine Learning* (Malysiak Jonathan & Jain Suresh, Eds.; 3rd ed.). Packt.

Shemin, R. J., Dziuban, S. W., Kaiser, L. R., Lowe, J. E., Nugent, W. C., Oz, M. C., Turney, D. A., & Wallace, J. K. (2002). Thoracic surgery workforce: snapshot at the end of the twentieth century and implications for the new millennium. *The Annals of Thoracic Surgery*, *73*(6), 2014–2032. https://doi.org/10.1016/S0003-4975(02)03585-3

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

Simian, D., Stoica, F., & Bărbulescu, A. (2020). Automatic optimized support vector regression for financial data prediction. *Neural Computing and Applications*, *32*(7), 2383–2396. https://doi.org/10.1007/s00521-019-04216-7

Sing, C., Love, P. E. D., & Tam, C. M. (2012). Stock-Flow Model for Forecasting Labor Supply. *Journal of Construction Engineering and Management*, *138*(6), 707–715. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000485

Smola, A. J., Sch¨olkopf, B., & Sch¨olkopf, S. (2004). A tutorial on support vector regression *. In *Statistics and Computing* (Vol. 14). Kluwer Academic Publishers.

Stitt-Gohdes, W. L., & Crews, T. B. (2004). The Delphi Technique: A Research Strategy for Career and Technical Education. *Journal of Career and Technical Education*. https://doi.org/10.21061/jcte.v20i2.636

Swindel, B. F. (1976). Good ridge estimators based on prior information. *Communications in Statistics-Theory and Methods*, *5*(11), 1065–1075.

Taylor, T. R. B., & Maloney, W. F. (2013). *Forecasting Highway Construction Staffing Requirements*. Transportation Research Board. https://doi.org/10.17226/22514

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Trawinski, B., Smetek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, *22*(4), 867–881. https://doi.org/10.2478/v10006-012-0064-z

Tsuji, S., Hase, T., Yachie-Kinoshita, A., Nishino, T., Ghosh, S., Kikuchi, M., Shimokawa, K., Aburatani, H., Kitano, H., & Tanaka, H. (2021). Artificial intelligence-based computational

framework for drug-target prioritization and inference of novel repositionable drugs for Alzheimer's disease. *Alzheimer's Research & Therapy*, *13*(1), 1–15.

Urgen Schmidhuber, J. ¨, Wierstra, D., Gagliolo, M., & Gomez, F. (n.d.). *Communicated by Yoshua Bengio Training Recurrent Networks by Evolino*. http://direct.mit.edu/neco/article-pdf/19/3/757/816836/neco.2007.19.3.757.pdf

Uszkoreit, J. (2017). Transformer: A novel neural network architecture for language understanding. *Google AI Blog*, *31*.

Vázquez-Ramos, R., Leahy, M., & Hernández, N. E. (2007). The Delphi Method in Rehabilitation Counseling Research. *Rehabilitation Counseling Bulletin*, *50*(2), 111–118. https://doi.org/10.1177/00343552070500020101

Wahid, A., Breslin, J. G., & Intizar, M. A. (2022). Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. *Applied Sciences*, *12*(9), 4221.

Wei, C., Chen, J., Song, Z., & Chen, C. I. (2019). Adaptive virtual sensors using SNPER for the localized construction and elastic net regularization in nonlinear processes. *Control Engineering Practice*, *83*, 129–140. https://doi.org/10.1016/j.conengprac.2018.10.017

Wong, J. M., Ng, S. T., Chan, A. P., & Chiang, Y. H. (2009). *FORECASTING CONSTRUCTION MANPOWER DEMAND: AN ECONOMETRIC MODEL*.

Wong, J. M. W., Chan, A. P. C., & Chiang, Y. H. (2007). Forecasting construction manpower demand: A vector error correction model. *Building and Environment*, *42*(8), 3030–3041. https://doi.org/10.1016/j.buildenv.2006.07.024

Wong, J. M. W., Chan, A. P. c., & Chiang, Y. H. (2011). Construction manpower demand forecasting: A comparative study of univariate time series, multiple regression and econometric modelling techniques. *Engineering, Construction and Architectural Management*, *18*(1), 7–29. https://doi.org/10.1108/09699981111098667

Wong, J. M. W., Kong, H., & Chiang, Y. H. (2004). A critical review of forecasting models to predict manpower demand. In *The Australian Journal of Construction Economics and Building* (Vol. 4, Issue 2).

Xiang, Y., & Song, H. (2020). Application of CNN-ATTBiLSTM Hybrid Algorithm in Natural Language Processing. *Journal of Vibration Testing and System Dynamics*, *4*(2), 163–171.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., & Sugiyama, M. (2014). High-Dimensional Feature Selection by Feature-Wise Kernelized Lasso. *Neural Computation*, *26*(1), 185–207. https://doi.org/10.1162/NECO_a_00537

Yang, W. J., & Kim, Y. S. (2019). Manpower Allocation Model for Construction Site Office Engineers based on Inherent Technical Risks. *KSCE Journal of Civil Engineering*, *23*(3), 947–957. https://doi.org/10.1007/s12205-019-0663-4

Yunpeng, L., Hou, D., Junpeng, B., & Yong, Q. (2017). *Multi-step Ahead Time Series Forecasting for Different Data Patterns Based on LSTM Recurrent Neural Network*. 305–310. https://doi.org/10.1109/WISA.2017.25

Zaheer, S., Anjum, N., Hussain, S., Algarni, A. D., Iqbal, J., Bourouis, S., & Ullah, S. S. (2023). A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. *Mathematics*, *11*(3), 590.

Zhang, Z., Zhao, Y., Canes, A., Steinberg, D., & Lyashevska, O. (2019). Predictive analytics with gradient boosting in clinical medicine. *Annals of Translational Medicine*, *7*(7), 152–152. https://doi.org/10.21037/atm.2019.03.29

Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, *28*(1), 162–169. https://doi.org/10.21629/JSEE.2017.01.18

Zhao, Y., Qi, K., Chan, A. P. C., Chiang, Y. H., & Siu, M. F. F. (2022). Manpower forecasting models in the construction industry: a systematic review. In *Engineering, Construction and Architectural Management* (Vol. 29, Issue 8, pp. 3137–3156). Emerald Group Holdings Ltd. https://doi.org/10.1108/ECAM-05-2020-0351

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429. https://doi.org/10.1198/016214506000000735

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Appendix

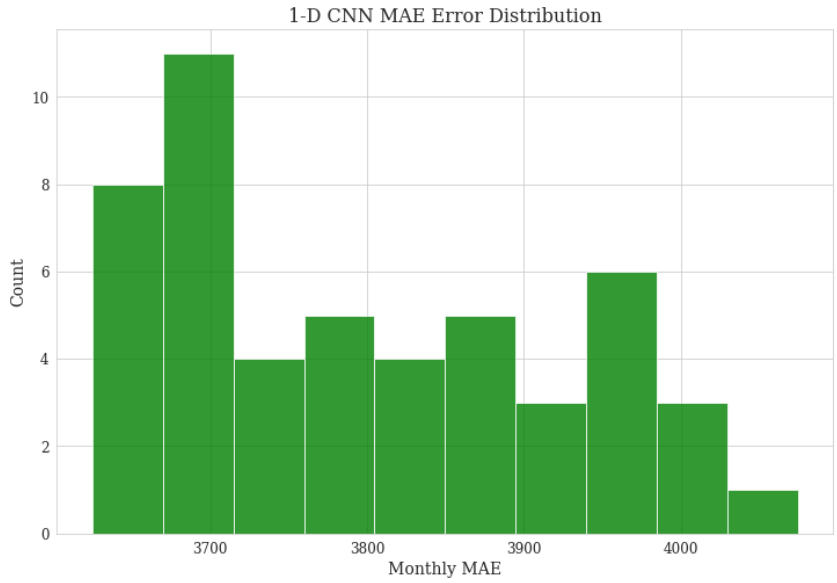

Figure 23: Short range 1-D CNN MAE distribution

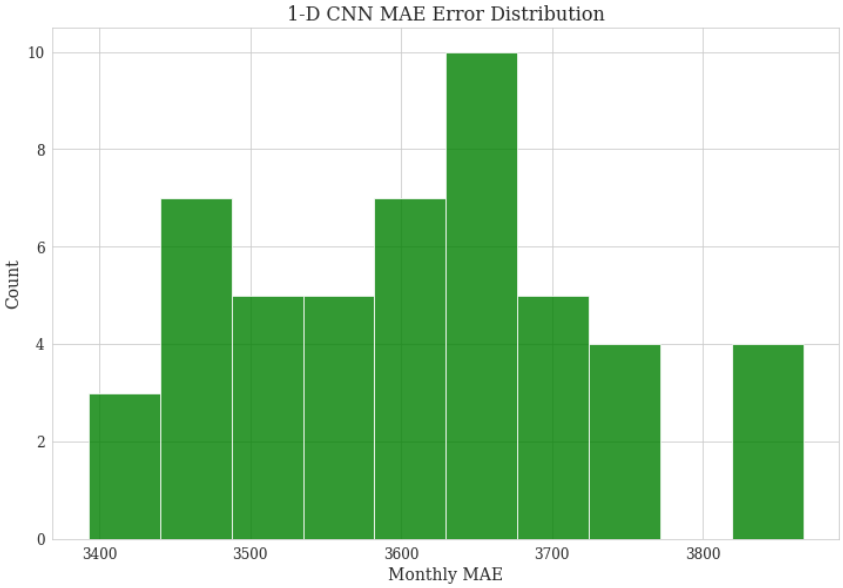

Figure 24: Long range 1-D CNN MAE distribution

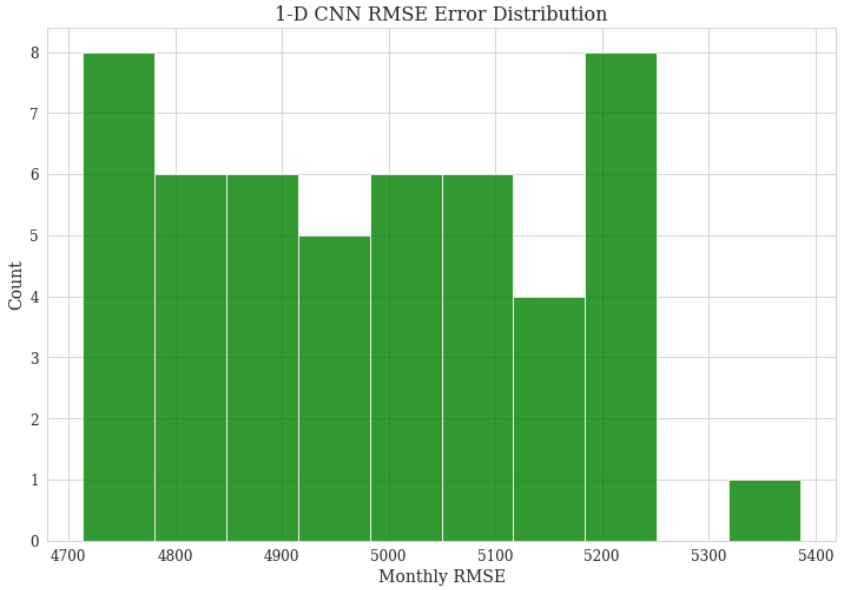

Figure 25: Short range 1-D CNN RMSE distribution

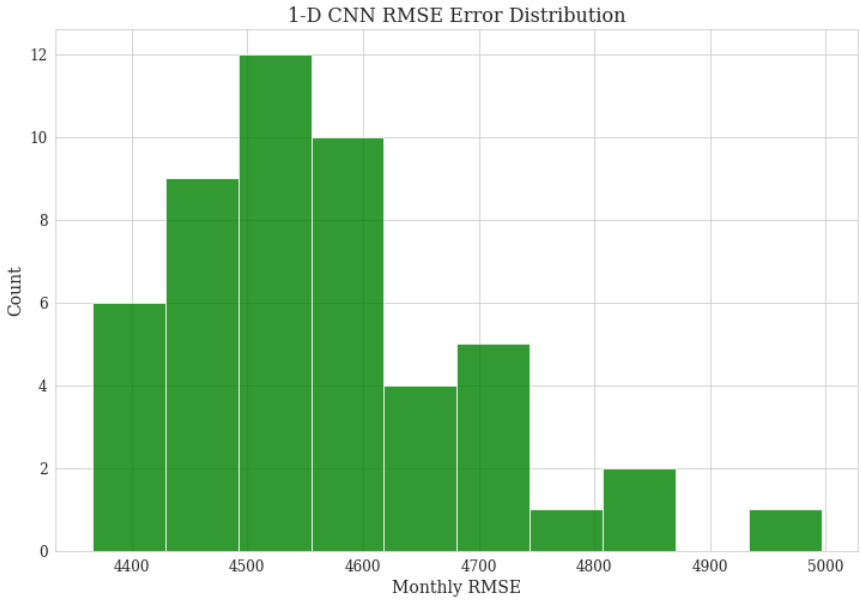

Figure 26: Long range 1-D CNN RMSE distribution

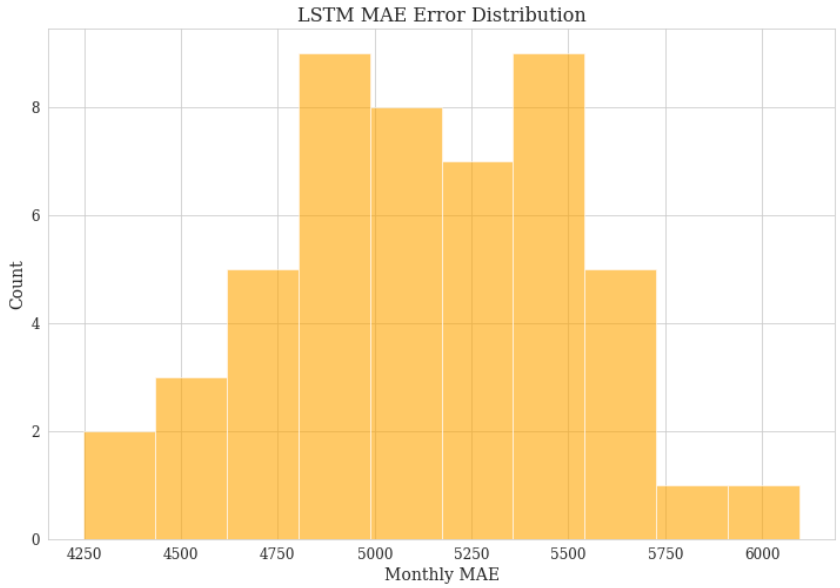

Figure 27: Short range LSTM MAE distribution

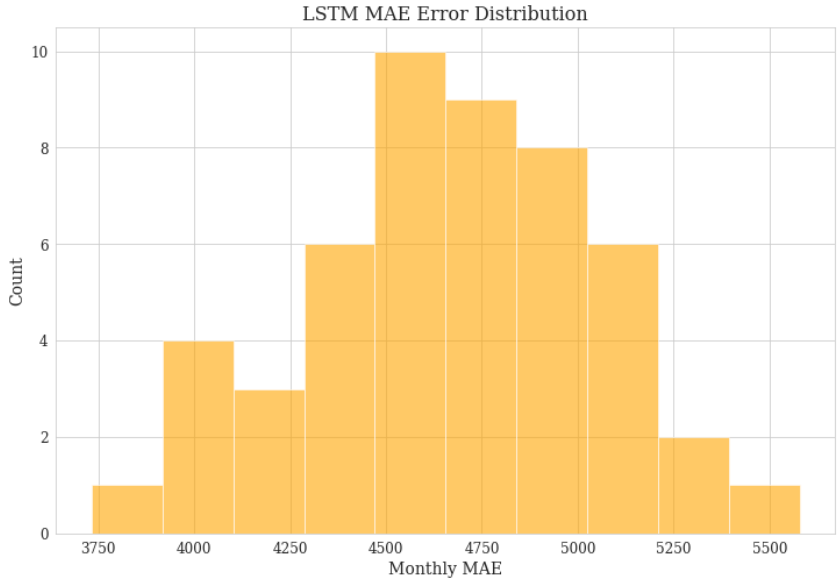

Figure 28: Long range LSTM MAE distribution

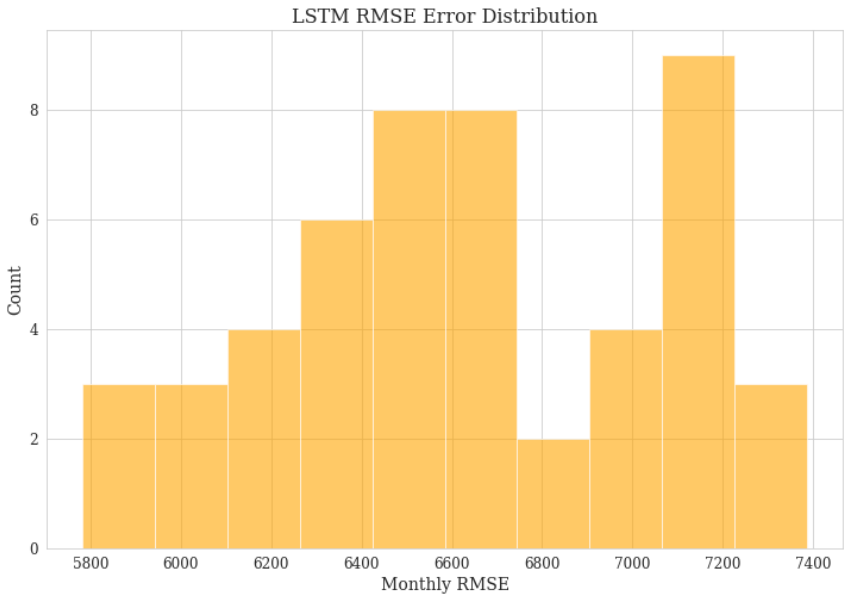

Figure 29: Short range LSTM RMSE distribution

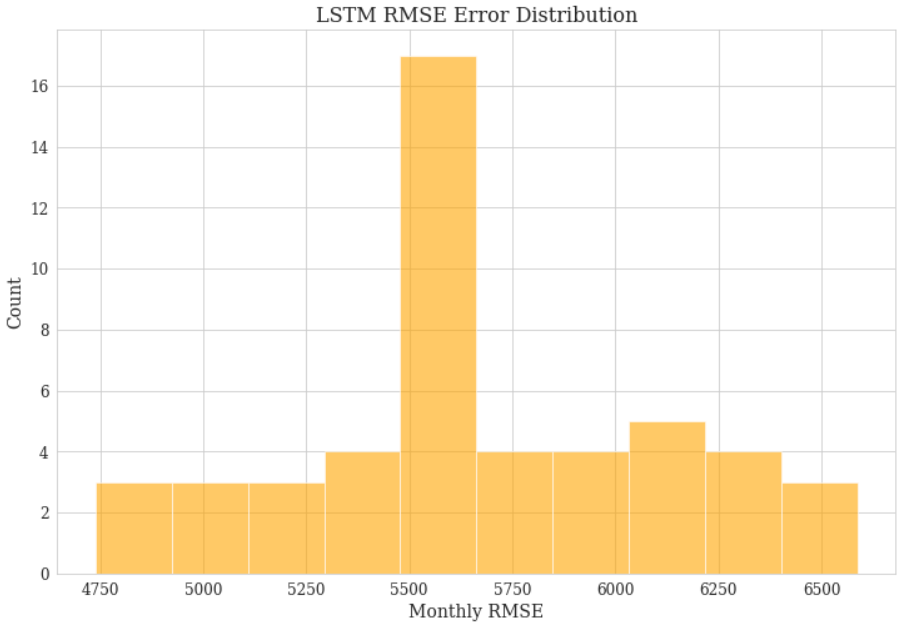

Figure 30: Long range LSTM RMSE distribution

Table 3: Hyperparameters for machine learning models

Model	Hyperparameter	Search space	Tuning technique
RF	Number of estimators Min. Sample Split Max. depth Max features	10 – 1000 2 – 6 3 – 10 sqrt, log2	Grid search
SVR	Kernel Degree Gamma Epsilon Max iteration	linear, poly, rbf, sig 2 – 5 Scale, auto 0.01 – 0.1 100 – 20000	Bayesian Opt.
GBR	Learning rate Number of estimators Criterion Min sample split Max. depth Alpha	0.001 – 0.1 100 – 10000 Friedman_mse, mse, squared error 0 – 0.5 1 – 10 0 – 0.5	Bayesian Opt.
DT	Criterion Max depth Min sample split Min sample leaf Max features Max leaf nodes	Squared error, friedman, poisson 3 – 50 2 – 10 1 – 100 Log2, sqrt 5 – 20	Grid Search
NN	Batch size Learning rate Optimizer Num of Layers Hidden layer size	32 – 128 0.001– 0.1 Adam, AdamW 2 – 12 16 – 256	Manual, Bayesian Opt.
RR	Alpha Solver	0 – 100 ‘auto’, ‘svd’, ‘lbfgs’, ‘lsqr’, ‘sag’, ‘saga’	Grid Search
EN	Alpha Max iteration L1 ratio	0 – 100 100 – 5000 0 – 1	Grid Search
LR	Alpha Max iteration	0 – 100 100 - 5000	Grid Search

Table 4: Arima model coefficient

	coef	std err	Z	P > |z|	[0.025	0.975]
ar.L1	0.4367	0.386	1.132	0.257	-0.319	1.193
ar.L2	0.5550	0.376	1.478	0.139	-0.181	1.291
ma.L1	0.0455	0.367	0.124	0.901	-0.675	0.766
ma.L2	-0.6015	0.156	-3.859	0.000	-0.907	-0.296
ar.S.L12	0.9706	0.126	7.710	0.000	0.724	1.217
ma.S.L12	-0.8593	0.340	-2.527	0.011	-1.526	-0.193
sigma2	8.998 e+07	5.94e-09	1.52e+16	0.000	9e+07	9e+07

Table 5: Average Model performance for project-level models

	ME	RMSE	MAE	MAPE	R^2
OLS Reg.	23489.6	2503.373	778.3936	0.079852	0.780971067
RidgeNet Reg.	11913.1	1710.173	683.967	0.073116	0.768877511
ElasticNet Reg.	11506.1	1677.375	672.1263	0.073462	0.766620213
Lasso Reg.	11506.1	1677.375	672.1263	0.073462	0.766620213
Gradient Boost Reg	11393.13	1647.381	673.4282	0.073779	0.774220435
CatBoost Reg.	12994.39	1752.818	697.4843	0.076279	0.758206156
Random Forest	**9063.574**	**1203.836**	**468.412**	**0.047472**	**0.907965659**
Decision Tree	11693.41	1703.9	710.1871	0.079046	0.745708195
Support Vector Reg.	11523.79	1661.664	673.4373	0.073399	0.770317195
Neural Nets.	11163.85	1609.595	659.4853	0.075067	0.799918486

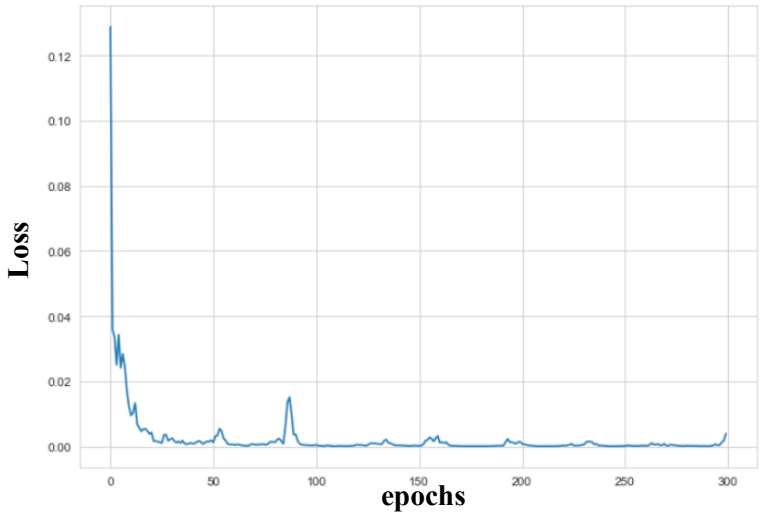

Figure 31: Sample loss per epoch (1-D CNN)

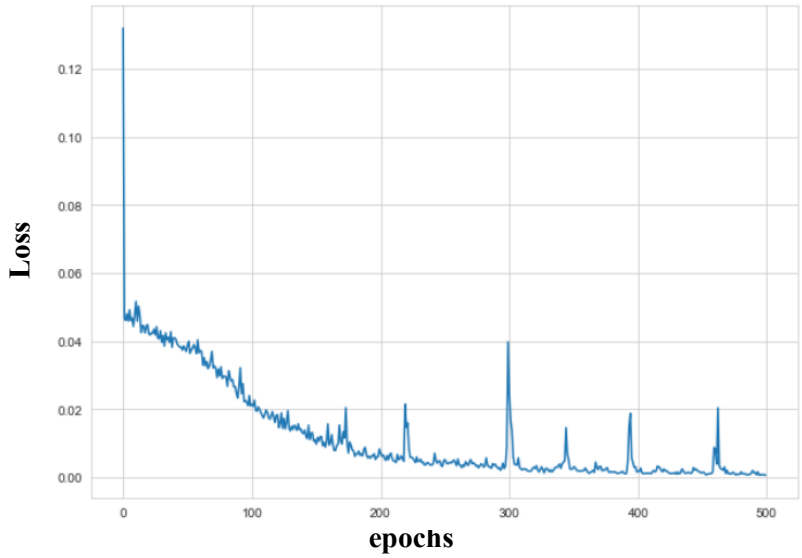

Figure 32: Sample loss per epoch (LSTM)