AMERICAN UNIVERSITY OF BEIRUT

FORECASTING FIELD PRODUCTION USING MACHINE LEARNING TIME SERIES

by
JOHNNY JOSEPH SAWMA AWAD

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Civil and Environmental Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2022

AMERICAN UNIVERSITY OF BEIRUT

FORECASTING FIELD PRODUCTION USING MACHINE LEARNING TIME SERIES

by

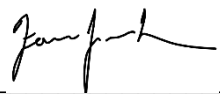
JOHNNY JOSEPH SAWMA AWAD

Approved by:

Signature

Dr. Issam Srour, Associate Professor — Advisor
Dept. of Civil and Environmental Engineering

Dr. Ibrahim Alameddine, Associate Professor — Member of Committee
Dept. of Civil and Environmental Engineering



Dr. F. Jordan Srour, Associate Professor — Member of Committee
Dept. of Information Technology and Operation Management
Adnan Kassar School of Business,
Lebanese American University

Date of thesis defense: September 12, 2022

AMERICAN UNIVERSITY OF BEIRUT

THESIS RELEASE FORM

Student Name: Sawma Awad_____Johnny_____Joseph_____
Last First Middle

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of my thesis; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes:

☐ As of the date of submission

☐ One year from the date of submission of my thesis.

☒ Two years from the date of submission of my thesis.

☐ Three years from the date of submission of my thesis.

____ _____September 12,2022

Signature Date

ACKNOWLEDGEMENTS

I would like to provide many thanks for my thesis advisor Dr. Issam Srour for his continuous support throughout this journey. Dr. Issam is an exceptional mentor who showed immense support, assistance, and guidance throughout this work. Being part of Dr. Issam's research team enhanced my research skills, analytical thinking skills, and professionalism skills.

My deep gratitude also goes to the committee members Dr. Ibrahim Alameddine and Dr. Jordan Srour for providing their time to be part of this thesis work, and for their insightful feedback throughout.

Also, I would like to thank Sena Assaf, who was part of Dr. Issam Srour research team, for assisting me to get started with this research work and for providing proper guidance.

I would also like to thank my parents, family, and friends for being a resilient support system throughout this journey.

ABSTRACT
OF THE THESIS OF

Johnny Joseph Sawma Awad for Master of Engineering
Major: Civil Engineering

Title: Forecasting Field Production Using Machine Learning Time Series

Accurate forecasting of field production is an essential ingredient for effective planning, management, and control of construction resources. Traditional forecasting methods often rely on historical data and on the subjective experience of project managers that fail to account for the dynamic and unique nature of construction operations. As such, this thesis presents a generic framework to collect, mine and analyze ongoing field production data. It incorporates developing a time series machine learning based model to forecast production for the upcoming days. Also, to ensure its performance, the model was benchmarked with the traditional statistical based Autoregressive Integrated Moving Average (ARIMA) time series model and the Multi-layer perceptron (MLP) neural network that does not include the time factor. The framework was tested on a case study of excavation activities of a real-world infrastructure project for eight months. The developed machine learning time series model produced a satisfactory performance to forecast field production, and it showed improvements over the statistical ARIMA time series model and the MLP model with no time lag. The proposed framework can assist project managers to accurately forecast field production in order to make data-driven resource allocation decisions.

TABLE OF CONTENTS

ILLUSTRATIONS

Figure

TABLES

Table

ABBREVIATIONS

MLP: Multi-layer Perceptron

ANN: Artificial Neural Networks

ARIMA: Autoregressive Integrated Moving Average

SARIMA: Seasonal Autoregressive Integrated Moving Average

SARIMAX: Seasonal Autoregressive Integrated Moving Average with Exogenous Factors

ML: Machine Learning

CHAPTER I

INTRODUCTION

Construction projects are dynamic in nature requiring continuous performance monitoring and schedule updates. Among various project performance metrics, productivity rates are considered as essential performance metrics that should be estimated and tracked, as they have a direct impact on project time and cost (Shehata & El-Gohary, 2012). Productivity is the index reflecting the construction output produced (quantity) versus input resources consumed. At the planning stage of a project, initial productivity estimates are established based on either subjective judgment, historical data, or published data such as the RSMeans (Song & AbouRizk, 2008). However, the accuracy of the initial estimates compared to the actual values is low because the estimates are prepared prior to the execution with limited information about the activities and contain various assumptions about the activities (Rao and Grobler 1997). Such deviation is unavoidable since the estimates cannot account for a project's dynamic and unique behavior.

Productivity is affected by several factors, including the site conditions, environmental, managerial, and operational aspects of a project (Hasan et al., 2018). During the construction process, productivity of resources fluctuates over time due to the impact of these various factors (Hwang & Liu, 2010). This continuous change is normal because a construction process involves dynamically interacting tasks (Halpin and Riggs 1997). However, project managers struggle to mitigate this fluctuation in a timely manner because they fail to perceive this change in performance over time. Consequently, schedule delays and cost overruns are detected at an advanced stage leading to overtime work and costly fixes. In addition, inefficiencies in managing resources are produced,

thereby posing a major impact on the technical and economic feasibility of the project (Kassem et al.,2021).

In general, the control methods used to manage a construction project are reactive. These methods normally compare actual outcomes to estimates, and corrective action is taken if comparisons show significant deviations (Russell et al. 1997). Due to the dynamic nature of projects, the reactive control method is suboptimal since proper resource management procedures cannot be taken in a timely manner. Therefore, a proactive approach, where project performance is continuously predicted along the project execution, should be adopted to effectively plan and control the ongoing operations (Hwang & Liu, 2010). This proactive project control is achieved by continuously analyzing the ongoing field productivity data and predicting future performance. However, in practice, project managers are not taking advantage of ongoing field data to accurately predict future performance. Instead, they commonly perform traditional forecasting methods that rely on historical data and on subjective experience, and these methods fail to account for the dynamic and unique nature of construction operations.

As such, it is essential to consider more advanced forecasting methods along with proactive planning approaches (Hwang & Liu, 2010). With the development of data analytics and machine learning (ML) tools, it has become possible for project managers to rely on predictive models to enhance the control and management of construction operations. Despite the presence of powerful predictive tools, project managers are not taking the advantage of analyzing ongoing field production data to extract valuable insights through the use of data analytical tools.

The project manager's tasks include managing resources (labor, materials, machinery) to meet schedule and budget requirements. The production output of these

various interacting resources can be derived from productivity rates. For on-site decision making, knowing the production output of the various interacting resources can also be useful to predict project performance instead of taking the productivity of one resource separately. This will lead project managers to compare their target production values with the estimated production values, and resource decision making would be established based on this comparison. For instance, excavation activities in large infrastructure projects require several interacting resources, and they tend to be dynamic in nature where the production of resources fluctuates due to internal factors (i.e.: operational, and managerial) and external factors (i.e.: environmental weather features and site conditions). During the construction process, the project manager is put under uncertainty due to the failure of properly anticipating the change in performance over time. As a result, inefficiencies in resource management are produced because the project manager fails to accurately estimate the production values of the various interacting resources, thereby leading to schedule delays and cost overruns.

Construction operations tend to be continuous in nature, and time dependency of performance between consecutive time step values might exist (Hwang & Liu,2010). An activity can span a long duration and can be continuous in nature, thereby instigating the need for project managers to reduce uncertainty in performance estimation and to improve the management of resources. To proactively control a project, a robust predictive model reflecting the changing behavior of performance over time is needed thereby leveraging ongoing field production data. Such modeling techniques include time series tools that are beneficial in extracting the underlying patterns of the ongoing data (Hyndman & Athanasopoulos, 2015) instead of relying on historical data obtained from previous projects. The aim of this study is to develop and illustrate a generic framework that

utilizes advanced machine learning time series analysis to accurately forecast the production output of resources over time using ongoing field data. By improving the prediction of the resources' production values, this framework acts as a decision-making tool to properly achieve data-driven resource management decisions to reduce time and cost impacts.

The remainder of this document includes a literature review, description of the study objectives and methodology, preliminary results, and remaining activities.

CHAPTER II

LITERATURE REVIEW

A. Productivity Measurement in Construction

Several metrics exist to properly monitor and control the performance of a construction project. Construction productivity has been gaining interest and is considered to be one of the most researched areas in the past few decades (Hasan et al., 2018). Productivity is mainly used to measure the efficiency or the rate of production of an activity. Various terms are used to describe productivity in the construction industry: production rate, performance factor, unit person-hour (p-h) rate, among others (Dozzi & AbouRizk, 1993). Productivity is usually computed as the ratio of output produced (e.g, cubic meter of concrete) to input resources consumed (e.g, labor and equipment hours). Talhouni (1990) presents three different productivity measures for construction: 1) single factor productivity (SFP), that measures the productivity of construction activities using only one resource input (labor), 2) multi factor productivity (MFP), that measures the productivity of construction activities using any combinations of the three resource inputs (i.e.: labor, material, and equipment); 3) total factor productivity (TFP), that measures the productivity of construction activities using only five resource inputs (i.e.: labor, material, equipment, capital, and energy). In construction, productivity is very commonly understood and computed as the SFP representing output produced per man-hour, presented in Eq (1).

$$\text{Productivity} = \frac{\text{Proudction output}}{\text{Labor hours}} \qquad \text{Eq.}(1)$$

Yet, productivity values can be represented in several ways where some contractors prefer to use the inverse of Eq. (1) (Shehata & El-Gohary, 2012). Also, productivity values can be reported for the whole crew instead of computing it per man-hour (Oral & Oral, 2010). For instance, the RSMeans, one of the world's leading construction estimating database, reports productivity as the daily output or daily production of a crew. The literature does not provide an absolute definition or formula for the productivity index, so it has many different forms with respect to the scope and nature of both the outputs and resources considered (Shehata & El-Gohary, 2012). For example, for the case of an excavator equipment, productivity can be reflected by either cycle time or by total earthwork of volume produced (Kassem et al.,2021). In this paper, the production output of the whole crew is the target variable to be forecasted by knowing the input resource hours that will be used.

Productivity values play a key role in the control and management of projects due to their direct impact on time and cost (Shehata & El-Gohary, 2012). This impact can be explained through Eq. (2) and Eq. (3).

$$\text{Duration (Hours)} = \text{Quantity of work} \,/(\text{Quantity/Hours}) \qquad \text{Eq. (2)}$$

and $$\text{Cost} = \text{Crew Cost} \times \text{Duration/Hours} \qquad \text{Eq. (3)}$$

In fact, productivity values have a major role in the project planning phase since they can be used to establish useful estimates that are used as a guideline along the project.

Research on construction productivity has either focused on the identification of the factors affecting construction productivity or on the development of predictive models for construction productivity.

B. Factors Affecting Productivity

Because of the dynamic and unique behavior of projects, construction productivity fluctuates overtime due to several factors (Hwang & Liu ,2010). Substantial research has been achieved to identify the relevant factors affecting construction productivity and to quantify the impact of such factors on productivity. Determining the relevant factors affecting construction productivity is an important criterion for improving productivity in the construction industry (Choy & Ruwanpura, 2005; Ghoddousi & Hosseini, 2012). In construction projects, both labor and equipment are essential resources to achieve a task. Construction activities can be categorized into two: labor-intensive activities, where labor is the main driver of productivity (e.g., mechanical and electrical activities) (Jarkas, 2010), and equipment-intensive activities, where equipment is the main driver of productivity (e.g., earthmoving activities) (Ok & Sinha, 2006). The factors that affect the productivity of labor-intensive activities and equipment-intensive activities are different because the resources and the productivity measures for these two types of construction activities are different (Seresht & Fayek, 2018). This section further tackles the factors that affect the labor productivity and equipment productivity; in addition, the learning curve and time factor is presented.

1. Labor Related Factors

Construction productivity is mainly dependent upon human effort and performance (Jarkas, 2010). Because construction is a labor-intensive industry (Alaghbari et al., 2019), prior research on the determination of the factors affecting activity-level productivity is mainly focused on labor-intensive activities (e.g. Tsehayae & Fayek, 2014; Hwang et al., 2016; Naoum, 2016). Construction productivity varies from one project to another due to the environmental and managerial conditions (Mirahadi & Zayed, 2016). Since construction projects are labor-intensive, various factors, such as weather, site and job conditions, labor skill, training and motivation, can impact construction productivity (Hasan et al,2018). Commonly analyzed factors include change orders (Hanna & Gunduz 2004; Moselhi & El-Rayes 2005), overtime (Hanna et al. 2005; Thomas & Raynar 1997), management practices (Pocock et al. 1996), and environmental components (Sanders & Thomas 1991; Sonmez & Rowings 1998). Through an extensive literature review on productivity in construction, Park (2006) identified that the scheduled overtime, change orders, materials management, weather and human factors are the main factors that influence the productivity rate. Recently, Hasan et al. (2018) conducted a systematic literature review of papers published between 1986 and 2016 on the identification of the factors influencing construction productivity. They critically examined the construction productivity literature by examining several characteristics of these studies, such as their countries of origins, the publication year, level of expertise of the survey respondents, and the most common factors between the different studies. As such, the study concluded that the most influential and common factors obstructing construction productivity, in the decreasing order of their criticality are: non-availability of materials, inadequate supervision, skill shortage, lack of proper tools and equipment,

incomplete drawing and specifications, poor communication, rework, poor site layout, adverse weather conditions and change orders (Hasan et al., 2018).

Although intensive investigations were performed concerning these factors, researchers have not approved on a universal set of factors that have major impact on labor productivity. Moreover, the preceding surveys and studies identifies that the most significant factors differ in each country and for each activity. In fact, Polat and Arditi (2005) showed that in developing countries the critical factors differ from those in developed countries. Therefore, these productivity factors are challenging to consider due to dynamic and unique nature of every project.

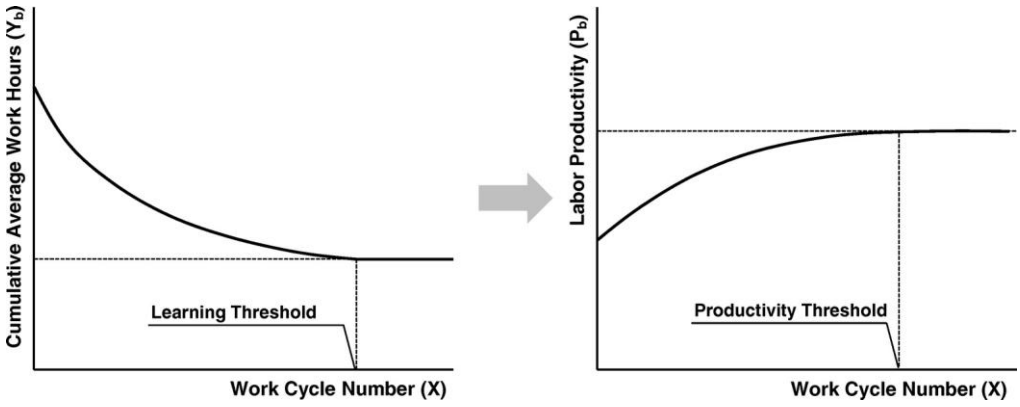
2. Equipment Related Factors

Construction equipment play an essential role in the modern construction industry. Several equipment-intensive activities are performed in various construction projects, including tunneling (Shaheen et al. 2009), pavement construction (Choi & Ryu 2015), earthmoving (Ok & Sinha 2006; Jabri & Zayed 2017), and pile construction (Zayed & Halpin 2005). Various studies state that the estimated equipment production rates gathered before the start of the project and the actual data from the field have substantial differences due to several factors (Rashidi et al.,2014). Construction equipment exist in various sizes, types, and functionality. Therefore, to improve equipment productivity, it is important to identify the specific factors affecting equipment's productivity. Several studies examined the productivity factors affecting the various types of equipment. For instance, site terrain attributes (e.g., soil type, soil conditions, gradient of slope, and haul surface), environmental features (e.g., temperature, precipitation, humidity, etc.), equipment characteristics (e.g., machine weight, capacity,

maintenance, swing angle, tractive force etc.), and the operator's skills are important factors that are considered to affect the performance of major earthmoving equipment (Shi & AbouRizk, 1998; Edwards & Holt 2000; Yang et al. 2003; Deshmukh & Mahatme, 2016).

3. Learning Curve and Time Factor

The time factor plays a critical role in construction since the learning curve of labor progresses along the execution of an activity. Repetitive construction operations instigate a phenomenon known as learning effect. The learning curve represents the improvement in work efficiency along the repetition of the same or similar task, thereby considering time as a factor. This curve is developed by plotting the time or cost required to complete one unit, which is a reciprocal expression of productivity of Eq(1), as a function of units produced.



(a) (b)

Figure 1 (a) Learning Curve over time and (b) Productivity Curve over time

Several models have been performed to explain learning curves for various activities (Everett & Farghal,1994; Gates & Scarpa, 1972;). Farghal and Everett (1997)

investigated the production time of repeated tasks using their learning curve model to predict remaining costs. In fact, the learning curves for construction activities tend to converge into a static value as repetition progresses. This convergence can be attainable under controlled conditions; however, in real operations, the convergence of a learning curve contradicts the fact that labor productivity fluctuates over time (Hwang & Liu, 2010). Hwang and Liu (2010) highlight the fact that the crew loses learning effect for several reasons during the period of operations in the long term. Therefore, productivity of a construction operation is more likely to fluctuate over time than for it to converge (Hwang & Liu, 2010).

C. Industry Practices for Estimating Productivity

Currently, project managers rely on either subjective judgment or historical project data to predict productivity in the project. Motwani et al. (1995) conducted a survey that showed that more than 20% of contractors rely on estimators' "gut feelings" and subjective opinions for most of their estimates. Consequently, the common practices in managing site equipment are highly dependent on the site managers' experience (Kassem et al.,2021). This subjective approach fails to account for the changing behavior of productivity and does not incorporate the effect of the influencing factors, thereby leading to inaccurate and unreliable productivity estimates. A more reliable and accurate estimate can be retrieved from historical project data because this data contains productivity information that can be used for future projects (Song & AbouRizk, 2008). However, this method fails to integrate the unique conditions of project complexity, the scope of work, and the environment of each project. As a result, these traditional methods

fail to reflect the dynamic and unique behavior of each project where productivity fluctuates over time due to internal and external factors.

In addition, project managers heavily focus on productivity prediction at the planning stage, and they rely on historical data excessively. Also, most studies in the literature have focused on the development of the initial productivity at the planning stage (Hwang & Liu, 2010). However, it is essential to continuously predict productivity along the execution of the project to mitigate the fluctuation in performance over time.

Also, only schedules and cost reports are constantly updated in most projects. Although contractors gather field productivity data, most contractors do not regularly update productivity data along the execution of the project (Hwang & Liu, 2010). Short-term meetings are performed to coordinate the schedule of work. However, for effective planning and control of a project, several performance metrics should be monitored and updated to reflect the current status of work and take necessary actions if required.

Furthermore, inefficiencies in managing resources are produced due to the unforeseen and fluctuating productivity along the project, and this poses a major impact on the technical and economic feasibility of the project (Kassem et al.,2021). The productivity value is a major indicator that drives the project manager to take resource management decisions. In order to control production output and manage input resources, it is essential to continuously understand, analyze, and predict the productivity of resources by relying on ongoing field data.

D. Data Driven Productivity Predictive Models

Researchers are now focusing on developing powerful predictive models that can estimate the productivity of resources in a project. Advanced tools such as artificial neural

network (ANN) and dynamic modeling techniques are being integrated to properly model the complex behavior of productivity.

In establishing a predictive model for productivity, the relation between factors affecting the productivity and the output cannot simply be pre-identified, so simple models like linear regression cannot guarantee significant results to forecast productivity (Mirahadi & Zayed, 2016). Nowadays, research is being directed to process non-linear data and establish non-linear models from field data (Tam et al.,2002). Therefore, more advanced models are needed to represent the complex and non-linear influence of factors on productivity. Consequently, Artificial Neural Networks (ANNs) and dynamic modeling techniques are showing great potential in estimating the productivity of resources. Several researchers were able to develop significant models using advanced tools to predict labor or equipment productivity.

The following section presents the predictive models that were developed with the various tools of neural networks, dynamic modeling, and time series.

1. Neural Network Models

Data analytics is the process of analyzing data to extract useful insights which can be utilized for decision making. Machine learning (ML) is the process where a computer program can learn from the training data and adapt to new unforeseen data without any human intervention. In fact, predictive models can be developed using ML algorithms. Predictive data analytics, a subcategory of data analytics, is the use of historical data to identify relationships and patterns to develop a predictive model. Artificial Intelligence (AI) systems are being considered to model complex relationships between the dependent and independent variables. ANN is a data-driven model that is capable of modeling the

non-linear and highly complex relationships by mapping the input stream of data to the output stream. ANN are ML algorithms that can learn from a training data set, analyze the complex patterns, and detect the corresponding relationships. To benefit from its advantages over the conventional methods, ANNs are being used in a wide variety of applications in construction management including cost estimation, risk analysis and safety, duration estimation, and especially productivity prediction (Kulkarni et al.,2017).

The following subsections present the ANN predictive models developed for labor productivity and the equipment productivity.

a. Labor Productivity ANN Predictive Models

Neural network architectures can be used to improve the automation in construction engineering and management (Moselhi et al., 1991). Various studies developed neural network models that were able to significantly improve the labor productivity predictions by incorporating the relevant influencing factors. Portas and AbouRizk (1997) developed a neural network model to estimate the construction productivity for the concrete formwork elements (i.e., slabs, walls, and columns). The system incorporates artificial neural networks, historical information, and input from experienced superintendents employed by a leading construction general contractor. Quantitative and qualitative factors were included in the model. The developed method was compared to an existing statistical model developed by the same contractor and was found to improve the quality of the estimates attained. Furthermore, AbouRizk et al. (2001) provided an approach based on ANNs that lets an estimator produce accurate labor production rates (labor/unit) for industrial construction tasks, such as welding and pipe installation. The factors were compiled from the literature, questionnaires, and telephones

from experienced superintendents. 33 factors were considered consisting of managerial and environmental input. The ANN model improved the quality of predictions over the current practice as well as over the use of a single-stage back-propagation network. Also, Ezeldin and Sharara (2006) designed a framework using neural networks to predict the productivity of forms assembly, steel fixing, and concrete pouring while including both quantitative and qualitative factors. The data was gathered from 18 experts and 92 survey responses, and the input includes managerial and environmental factors. They concluded that the networks adequately converged and have reasonable generalizing capabilities. In addition, Song and AbouRizk (2008) successfully applied ANN and discrete-event simulation in developing a labor productivity model for steel drafting and shop fabrication activities. They found that ANN, and generally simulation techniques, are effective in modeling individual activities that have complex detailed operations and a complex mapping relationship between productivity and influencing factors. Consequently, Oral and Oral (2010) utilized self-organizing map (SOM) to develop an ANN model-based algorithm with the aim to both analyze the relationship between various factors and to predict crew productivity under given conditions. Data was collected randomly from a construction site in Turkey, consisting of daily activities and observations of 144 concrete pouring, 101 formwork and 101 reinforcement crews. They concluded that SOM can predict productivity better than regression methods because of its complexity. Oral et al. (2012) showed that SOM is superior to Feed Forward Back Propagation (FFBP) and General Regression in predicting plastering crew productivity due its complexity in analyzing the data. Gerek et al. (2015) studied the performance of the feed forward neural network (FFNN) compared with radial basis neural network (RBNN) in modelling the productivity of masonry crews. The data was obtained from

several construction sites in turkey that include masonry crew data consisting of 147 crews. A variety of input factors were incorporated and analyzed, including labor related factors (skill, wages, overtime, age), contract-based factors (wage type, daily unit cost), and technical factors (wall type, mortar type). The results indicated that the neural computing techniques could be successfully employed in modeling crew productivity. Based on the MAPE values obtained for the models, the RBNN technique was found to be better than the FFNN technique. Mirahadi and Zayed (2016) developed a predictive model that integrates Neural-Network-Driven Fuzzy Reasoning to improve the prediction accuracy. They modeled the effect of the qualitative as well as the quantitative variables on construction productivity, including temperature, wind, humidity, precipitation, work type, crew size, and floor level. The developed model was able to successfully use historical data to forecast productivity of construction operations with a level of accuracy greater that what could be offered by traditional techniques. Heravi and Eslamdoost (2015) developed an artificial neural network model to measure and predict labor productivity in developing country of Iran. They investigated the influential factors on labor productivity using the Bayesian regularization and early stopping methods. Their work involved installing the concrete foundations of gas, steam, and combined cycle power plant construction projects. They considered 15 important factors related to the motivation of labor, supervision sufficiency, and competency. The following approach provided an enhanced method to model labor productivity. Consequently, El-Gohary et al. (2017) developed a framework to document, control, and predict contractor labor productivity utilizing ANN. They accounted for factors at the micro and macro levels of the project, and they applied the models to construction crafts, carpentry, and fixing reinforcing steel bars. The results for the ANN model showed an adequate convergence

with reasonable generalization capabilities. Table 1 shows the summary of the ANN predictive models of labor productivity.

Table 1 Summary of previous work to developing ANN labor productivity predictive model

Reference	Data Source	Methods	Productivity Factors included	Contribution
(Portas & AbourRizk, 1997)	Historical information and input from experienced superintendents	ANN MLP	Activity performance, Activity staff, Activity crew, Activity design, Activity dimensions, Activity repetition, Activity working conditions, Project complexity, Project structure, Project size, Project location, Project site	The ANN was compared to an existing statistical model developed by the same contractor and was found to improve the quality of the estimates attained.
(AbourRizk et al., 2001)	Historical data, literature, questionnaires, and telephones	ANN MLP	33 factors under nine groupings including general project characteristics, site, labor, equipment, overall project difficulty, general activity conditions, quantity, design, and activity difficulty.	The ANN model improved the quality of predictions of labor production rates (labor/unit) for industrial construction over the current practice as well as over the use of a single-stage back-propagation network.
(Ezeldin & Sharara, 2006)	Experts opinions and survey responses	ANN MLP	structural element under consideration; quantity of the element; crew size; labor skills; overtime; task complexity; material accessibility; degree of repetition; supervision; temperature conditions	The developed framework of neural networks indicates adequate convergence and relatively strong generalization capabilities to estimate labor's productivity in concrete related construction.
(Song & AbouRizk, 2008)	Historical data from a Project	ANN and simulation	project type, work scope, contract type, piece cloning, dynamic structure, fireproofing, dynamic structure, complexity, draftsperson qualification, crew size, client, engineering firm, engineering standards, administration, overtime, subcontract, total work quantity	ANN and simulation are effective in modeling individual activities that have complex detailed operations and a complex mapping relationship between productivity and influencing factors.
(Oral & Oral,2010)	Data from 144 concrete pouring, 101 formwork and 101 reinforcement crews	ANN Self-organizing Maps (SOM)	18 factors under 3 groupings labor related factors; work related factors; site management factors	SOM are satisfactory in clustering the data and prediction performance of the model is superior to similar artificial neural network models to predict concrete crew labor productivity.

(Gerek et al., 2015)	Historical data: 147 crews	Feed forward Neural Network (FFNN) **vs** radial basis neural network (RBNN)	13 factors under 3 groupings labor related factors; contract-based factors; and technical factors	The RBNN technique was found to be better than the FFNN technique in estimating masonry crew productivity.
(Heravi & Eslamdoost, 2015)	closed-ended questionnaire and historical data from power plant construction projects	ANN MLP	15 important factors related to the motivation of labor, supervision sufficiency, and competency	The influential factors on labor productivity and developing an artificial neural network to measure and predict labor productivity in developing countries using the Bayesian regularization and early stopping methods. This approach provides insight into better ways of modeling labor productivity.
(Mirahadi & Zayed, 2016)	Historical data	Simulation and Neural-Network-Driven Fuzzy Reasoning (NNDFR)	temperature, wind, humidity, precipitation, work type, crew size, and floor level.	The developed model helps researchers and practitioners use historical data to forecast the productivity of construction operations with a level of accuracy greater than what could be offered by traditional techniques. The main contribution of the proposed research was to promote the NNDFR model by fine-tuning its parameters and embedded techniques.
(El-Gohary et al., 2017)	Historical data computerized logs	ANN and simulation	micro level (project management and administration) and the micro/micro level (activity level at construction site)	This study contributes to the construction engineering and management body of knowledge by providing insight into using different ANN activation and transfer functions along with a wide range of influencing factors to benchmark the contractor's construction labor productivity.

b. Equipment Productivity ANN Predictive Models

Several researchers focused on developing ANN models to predict equipment productivity by incorporating the relevant environmental, managerial, operational, and equipment characteristic factors. These studies were able to improve the overall equipment productivity prediction estimates by using advanced neural network models. For instance, Chao and Skibniewsky (1994) developed a neural network model to estimate excavation-hauling operations productivity, represented by cycle times. The data was gathered from a batch program using the discrete event simulation method. The cycle time for the excavator was estimated using the trained neural network by inputting the swing angle, horizontal reach, vertical position, and soil type. The results demonstrated that the neural network approach can provide an accurate estimate and can be an efficient tool for construction productivity estimation.

Shi (1999) developed a neural network (NN) based approach for predicting the performance of earthmoving operations. The insufficiency of the industry data forced the author to rely on simulation. The model included several factors, including capacity of trucks, number of trucks, mean loading time, variance of loading time, mean hauling cycle time, variance of hauling cycle time, and weather conditions. Shi (1999) concluded that this neural network model is able to reliably assist the user in selecting the number of trucks to minimize the unit cost or in predicting the performance under a given crew.

Edwards and Holt (2000) developed a 'deterministic' multiple regression model to estimate the cycle time of excavators, as a measure of productivity. They collected the data from manufacturers' performance handbooks, and three variables were identified as accurate input predictors of the cycle time output: machine weight, digging depth and machine swing angle. Their model was able to successfully predict reliable cycle times

of excavators. As a result, Edwards and Griffiths (2000) aimed to improve the prediction accuracy of the multiple regression of Edwards and Holt. Therefore, they proposed a feed-forward ANN with backpropagation training to predict the hydraulic cycle time of excavators based on the data from manufacturers' performance handbooks. To make a reasonable comparison, they adopted the same input factors (i.e.: machine weight, digging depth and machine swing angle) of Edward and Holt model as predictors for the cycle time. They were able to reduce the error by 14% compared with the equivalent multiple regression equation. In addition, Tam et al. (2002) developed an ANN by using the same data set of Edwards and Holt (2000), and they compared the performance of their ANN with the multiple regression model of Edwards and Holt (2000). The input factors were grouped into two categories: Controllable factors (the digging depth in relation to the machine's maximum depth capacity, the swing angle; and capacity of the machine), and uncontrollable factors (the relative positions between the machine and the materials to be dug, the soil condition, the operator's skill, and site obstructions). The results showed that the use of the ANN model for predicting excavator cycling times can reduce uncertainty and increase accuracy. They concluded that the predictive performance of the ANN is superior to the multiple regression models. Ok and Sinha (2006) aimed to evaluate the potential of using non-linear network analysis for productivity estimation modeling. Therefore, they compared a transformed regression analysis and a nonlinear neural network model for dozer daily productivity estimation. The data was compiled from various projects of a contractor in North America, and the incorporated factors in the model as inputs are the equipment power, blade type, soil type, operator's skill, weather, grade, and hauling distance. Since the neural network model provided more accurate results than the regression model, they concluded that the

nonlinear ANN has the potential to improve the equipment productivity estimation model.

Schabowicz and Hola (2010) combined queuing theory with artificial neural networks (ANN) to predict the productivity of the earthwork machinery system consisting of excavators and haulers with varying machinery sizes and hauling distance. The sample consisted of 200 data points which were used to train (170) and test (30) the algorithm. Due to the lack of available real data, each neural network was generated by computer simulation. They considered several factors as input features, including number of excavators, number of trucks, excavator bucket capacity, truck loading platform, type of road surface. The results showed the effectiveness of the ANN for predicting the productivity of the collaborating earthmoving machines.

Kassem et al. (2021) developed a Deep Neural Network (DNN) model for estimating the productivity of excavators as a function of the daily excavated earth volume and established a benchmark for their productivity measure. The data was retrieved for each excavator from telematics systems installed on the equipment in the field. The obtained telematics data was used as input factors to predict the daily excavated earth volume per excavator, and these inputs include the fuel rate, vehicle weight, bucket volume, total fuel consumed, engine on time, travel time, digging, arm swing, and non-operating time. The predictive model was accurate resulting in a coefficient of determination (R2) and a Weighted Absolute Percentage Error (WAPE) of 0.87 and 69.64%, respectively. Table 2 shows the summary of the ANN predictive models of equipment productivity.

Table 2 Summary of previous work to developing ANN equipment productivity predictive models

Reference	Data Source	Methods	Productivity Factors included	Contribution
(Chao & Skibniewsky, 1994)	Batch program using the discrete event simulation method	ANN MLP	Excavator swing angle, horizontal reach, vertical position, and soil type	The results demonstrated that the neural network approach can provide an accurate estimate for excavator productivity and can be an efficient tool for construction productivity estimation.
(Shi, 1999)	Simulation	ANN MLP	Capacity of trucks, Number of trucks, mean loading time, Variance of loading time, Mean hauling cycle time, Variance of hauling cycle time, and Weather conditions.	The study demonstrated how ANN models are able to assist the user in selecting the number of trucks to minimize the unit cost or in predicting the performance under a given crew.
(Edwards & Griffiths, 2000)	Manufacturers' performance handbooks	ANN MLP	machine weight, digging depth and machine swing angle	The model provides an accurate prediction of cycle times for excavation works with any hydraulic excavator under given operating parameters. The ANN model proved to be more accurate in predicting excavator cycle time compared with the prediction performance of multiple regression.

(Tam et al., 2002)	Manufacturers' performance handbooks	ANN MLP	Controllable factors (digging depth in relation to the machine's maximum depth capacity, swing angle; and capacity of the machine), and uncontrollable factors (relative positions between the machine and the materials to be dug, soil condition, operator's skill, and site obstructions).	The study showed that the use of the ANN model for predicting excavator cycling times can reduce uncertainty and increase accuracy. The predictive performance of the ANN is superior to the multiple regression models.
(Ok & Sinha, 2006)	Historical data from Contractor	ANN MLP	equipment power, blade type, soil type, operator's skill, weather, grade, and hauling distance	The nonlinear ANN has the potential to improve the equipment productivity estimation model.
(Schabowicz & Hola, 2010)	Simulation	Queuing theory with ANN	number of excavators, number of trucks, excavator bucket capacity, truck loading platform, type of road surface	The study demonstrated the effectiveness of the ANN with combined queuing theory for predicting the productivity of the collaborating earthmoving machines.
(Kassem et al., 2021)	Telematics systems installed on the equipment in the field	Deep Neural Network (DNN)	fuel rate, vehicle weight, bucket volume, total fuel consumed, engine on time, travel time, digging, arm swing, not operating time.	The study provided an improved way of measuring productivity of excavators through telematics system and developed a DNN model for benchmarking their performance.

2. Dynamic Models

The existing predictive models of construction productivity are commonly developed using static modeling techniques that do not incorporate the changing behavior of projects over time, such as the ANN model developed by Heravi and Eslamdoost (2015). These static models rely on inputting a corresponding factor to get an output at a specific time period without considering the time sequence or time factor of the activity. However, dynamic modeling techniques can be more appropriate for construction productivity modeling since they can capture the changing behavior of construction productivity over time (Seresht & Fayek, 2018). A dynamic model represents the behavior of a system over time; moreover, it can capture the dynamic or changing behavior of construction projects and can determine the relationships which exist between the various factors influencing construction productivity (Seresht & Fayek, 2018).

The factors impacting construction productivity are mostly dependent on another, so variations in some factors can influence other factors (Mawdesley & Al-Jibouri, 2009). As a result, researchers tried to model the cause and effect relationships between the factors influencing construction productivity in addition to their individual impacts on productivity. These researchers integrated dynamic modeling methods to model the changing behavior of construction productivity over time. System dynamics is a simulation technique adopted to study and understand how systems change over time. System dynamics (SD) models were developed to model the behavior of construction productivity in order to assess and determine the influencing factors affecting productivity (Mawdesley & Al-Jibouri (2009); Nasirzadeh & Nojedehi 2013). Consequently, Nojedehi and Nasirzadeh (2017) developed an integrated fuzzy System Dynamics (FSD) approach for modeling and improving the construction labor

productivity. It is a system consisting of fuzzy logic and system dynamics. The fuzzy logic component accounts for the subjective uncertainty of factors, while the SD component captures the changing behavior of construction productivity and the relationships between factors affecting productivity. Several factors were considered and were divided into 5 main groups: motivational factors, work condition factors, site condition factors, workforce factors, and project management factors. Once the root causes of a decrease in labor productivity are identified through the SD model, various solutions are defined to improve the labor productivity. Accordingly, the various solutions are simulated using the FSD model. After applying a case study, they concluded that the proposed fuzzy-SD methodology offers a powerful simulation tool to improve construction labor productivity.

Seresht and Fayek (2018) developed a predictive model using the FSD technique to predict the multi-factor productivity of equipment-intensive activities considering three input resources of these activities (i.e., labor, equipment, and material). 72 crew-level and activity-level factors were identified through the literature review and were verified by expert knowledge. Through feature selection, 25 factors were picked and categorized into 5 groups: equipment-related factors, location-related factors, weather-related factors, task-related factors, crew-related factors, and material-related factors. The application of FSD technique contributes in developing an approach to reduce the uncertainty overestimation in the simulation results of FSD models in construction research. As a result, practitioners can accurately evaluate the actual system output, which is the actual productivity, according to the simulation results. Since FSD is still inadequate because of its low accuracy for modeling the complex relationships between the system variables, Seresht and Fayek (2020) developed a predictive model using neuro-fuzzy system

dynamic (n-FSD). The n-FSD combines ANN with the fuzzy system dynamic to predict the behavior of construction systems. The ANN (neuro) component is responsible to learn from the data by capturing the non-linear behavior. The n-FSD was tested through a case study by modeling earthmoving operations and was compared to the results of the FSD technique. The case study yielded a mean absolute percentage error (MAPE) of 16.92% and 55.96% for n-FSD models and FSD models, respectively. Consequently, the n-FSD improved the accuracy of FSD models in construction applications based on this comparative study. However, this model has a high computational cost compared with other techniques of SD and FSD (Seresht & Fayek, 2020).

It is significant that the ANN remains an effective tool in modeling construction productivity. Even for dynamic modeling, ANN was integrated with system dynamics and proved to be an effective tool in improving the prediction accuracy. Although the dynamic modeling techniques of system dynamics are important and seem to be promising, they are still not as practical and common as ANN models when it comes to forecasting of productivity. These dynamic models have a high computational cost and considered to be more complicated than other methods (Seresht & Fayek, 2020). Therefore, a more practical and effective ANN model that is able to capture the changing behavior of productivity over time is required.

3. Time Series Models

Time series analysis and forecasting can help model the changing behavior of productivity in construction since it accounts for the time dependencies of the data (Hwang & Liu, 2010). Time series analysis is a well-known method which has been utilized successfully in many fields for forecasting purposes (Hyndman &

Athanasopoulos, 2015). It consists of both finding patterns in the time series data and forecasting future values. Time series is periodic data where dependency among the data is affected by time. The patterns that may be detected in a time series are trends, cycles, seasonality, and residuals (Hyndman & Athanasopoulos, 2015).

Few studies have been previously made to use time series analysis in the construction management field (Hwang & Liu, 2010; Sparkes & McHugh 1984). Abdelhamid and Everett (1999) applied time series analysis to evaluate hoisting performance of a crane by comparing the CRANIUM technology with the traditional technique. The performed time series analysis analyzed the time consumption for shifting tools and materials by a 50-t mobile crane. The experiments were conducted in a controlled environment. This study showed the potential of time series analysis for analyzing experimental data to improve construction productivity.

Hwang and Liu (2010) investigated the performance of statistical time series forecasting in predicting short-term labor productivity. They incorporated time series for productivity prediction due to the belief that periodic data related to ongoing operations can provide an enhanced understanding about the changing behavior of trend in productivity. In this study, the time dependency between consecutive productivity observations was significant. The periodic weekly productivity data were collected from the labor-intensive work in construction—reinforced concrete work for building construction, including three activities: forming, concrete pouring, and concrete finishing. The data consisted of 24 sample series obtained from multiple building projects buildings for the same contractor, and it consists of 44 weekly productivity values (SF/ManHour). The overtime and weather factors (temperature, relative humidity, and wind speed) were included in the model. The study incorporates 5 statistical methodologies, including three variations

of the smoothing techniques-cumulative average, simple moving average (SMA), and simple exponential smoothing (SES)- and two variations of the time series analysis-Autoregressive moving average (ARIMA), and multivariate autoregressive (VAR). The one-step-ahead forecasts and errors of each model were generated to examine the validity of the selected models. Although results were not significant, the models were beneficial when there was a large gap between the budget and the actual productivity. Therefore, they concluded that the models are recommended when a high degree of uncertainty is present or expected. Consequently, Hwang (2010) proceeded to perform a validation for Hwang and Liu (2010) by comparing the performance of the five statistical time series forecasting models. Models were constructed using the first 30 observations, and productivity was predicted up to 4 weeks ahead from three sequential time origins. The performance was measured by evaluating the residual sum of squares and the correlation coefficients. Hwang (2010) concluded that univariate time series analysis was found to be the best-performing technique for this case.

E. Literature Gaps

When it comes to productivity prediction, the existing industry traditional methods of using subjective judgment and historical data are not always reliable and fail to capture the dynamic and changing nature of projects. Currently, project managers are not equipped with advanced data analytical tools and are not taking advantage of the ongoing field data to extract useful insights that could improve their decision making.

From the literature, it is evident that data analytics and ML tools, specifically ANN, can significantly assist project managers in predicting productivity in construction to mitigate time and cost overruns. The previously described labor and equipment ANN productivity predictive models were static models that do not integrate the time factor

and the changing behavior over time. Also, most models in this category rely on historical data from previous projects or simulation techniques, thereby failing to account for the unique nature of each project. Consequently, because productivity changes over time in a project, other studies focused on modeling the dynamic behavior of projects. Although the dynamic approaches seem to be promising, these dynamic modeling techniques are limited in accuracy, require incorporating several influencing factors, and can be computationally expensive.

Therefore, due to the dynamic and unique nature of construction projects, project managers need a data-driven and practical tool that can capture the changing behavior of performance over time by integrating time as a factor and extracting insights from ongoing field data. One study focused on developing a predictive model that integrates the time factor by using statistical time series analysis and forecasting to model the changing behavior of productivity.

Several statistical time series forecasting models exist. Among these is the well-known Autoregressive Integrated Moving Average (ARIMA) method that was utilized successfully in various applications (Hyndman & Athanasopoulos, 2015). However, its statistical properties limit its accuracy because it considers that the time series is produced from linear processes, so it might be inappropriate for real-world problems that are nonlinear (Khashei et al., 2009). Nonlinear and time-variant problems need to be tackled with tools that can model the complex behavior that changes with time. ANN time series modelling can be adopted in this case due to their substantial predictive capability which was demonstrated to outperform statistical time series methods utilized for the same objective (Laptev et al., 2017; Bianchi et al., 2017; Yunpeng et al., 2017).

Despite Hwang and Liu's (2010) attempt to develop a statistical time series model to predict labor construction productivity, the literature lacks a proper and effective framework that aims to forecast field production output using advanced ANN predictive time series models for the purpose of reducing uncertainty in production estimation and improving the field resource management decisions. Thus, advanced ANN time series models, other than statistical time series models, are required to further investigate the time factor relevance in predicting performance along the execution of the project. The construction operations tend to be continuous in nature, and time dependency of performance between consecutive time step values might exist (Hwang & Liu,2010). The time series model can extract the underlying patterns (trend, seasonality, cycles ,and residuals) that are exhibited along the construction process by accounting for the previous performance and the sequential behavior of the activity. The ANN time series models can properly model the complex non-linear behavior of construction operations by incorporating the time sequence of the activity as well as other internal and external factors to improve the prediction accuracy.

CHAPTER III

RESEARCH OBJECTIVE

It is evident that performance in construction fluctuates along the construction process due to several factors, and project managers are not able to anticipate this changing behavior over time. As a result, inefficiencies in resource management are produced, leading to time and cost overruns.

The objective of this study is to propose and illustrate a data-driven framework to forecast field production based on project internal and external factors using an advanced ANN time series predictive model. The model, which relies on ongoing field data, aims to reduce uncertainty in the fluctuating performance by improving the activity production estimation during the construction of a project. This tool can also assist project managers in their daily field decision-making to proactively plan their resources along the execution of the project. More specifically, the aim of the developed tool is to forecast production value per unit of time. The forecast will account for the external factors such as the prevailing weather conditions, in addition to internal factors such as the available equipment resource hours, and the previous production values. The time series model can implicitly capture the changing behavior of the performance by accounting for the sequential behavior of an activity. This is achieved by incorporating the time factor where patterns (e.g., trend, seasonality, cycles, and residuals) of the performance are accounted for at various phases.

This research study presents a generic framework to collect, analyze and restructure data of ongoing construction field activities. The obtained data is used to develop an ANN time series forecasting model to accurately predict production over time by accounting for the influencing factors. Also, to ensure that the developed forecasting models are

performing well, this research work presents two benchmarks model: 1) a statistical time series predictive model, 2) a neural network model with no time factor model. The proposed framework was tested on the excavation activities of a real-world infrastructure project for a period of eight months.

CHAPTER IV

METHODOLOGY

The methodology of this research work is divided into two parts. The first part presents a Machine Learning (ML) ANN time series generic framework that assists project managers in the execution of an activity to analyze the field production while accurately forecasting the production output. Also, it includes establishing a resource management plan to proactively plan the resources needed to achieve the corresponding performance.

The second part encompasses developing two benchmark models: 1) a statistical based time series model to analyze and forecast field production based on internal and external factors, and 2) a Multi-layer perceptron (MLP) model that does not incorporate the time factor (i.e.: time lag). The results of the two benchmark models will be compared with the results of the ML time series model. The MLP benchmark model is established by randomly shuffling the data points and not accounting for the time dependency in the data. The purpose of the benchmark MLP model is to compare its performance with the ML time series generic framework' s performance to show the importance of including the time dependency in the data.

A. Proposed Generic Framework to Estimate Field Production Using ANN Time Series

This section explains the steps followed to develop the ANN time series model. A multi-step framework is presented which includes data collection, data preparation and pre-processing, building the ANN time series models, forecasting application

implementation, and updating the model. The flowchart depicted in Figure 2 summarizes the proposed multi-step generic framework.

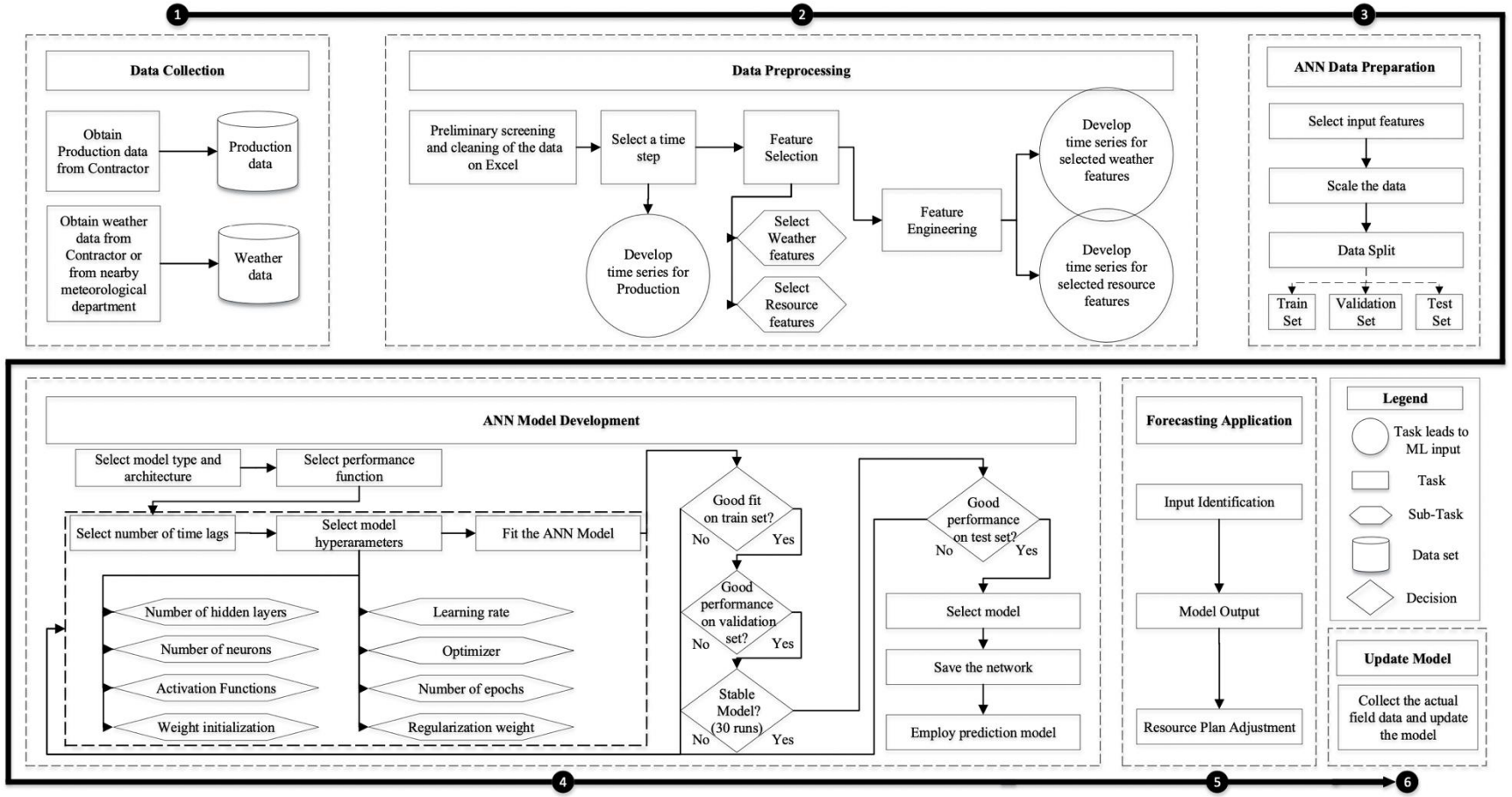


Figure 2 Generic Framework

1. Data Collection

Quality data are important to achieve a reliable forecast, which is an essential element of project controls. In construction, some activities are repetitive and continuous in nature spanning a long duration. For instance, excavation works can span approximately 2 years in a large infrastructure project. Thus, the historical and ongoing production data must be obtained from the contractor's database for a continuous and progressing activity of a specific project. The data can be extracted from daily report logs which include the daily production output with the corresponding information for the resources used (e.g., resource type and hours). Data must also be obtained for the factors that affect the activity's performance. Since construction activities are mainly performed in an external environment, weather data could also be a beneficial predictor for the activity's performance. The weather data could be obtained from the Contractor, if they exist, or from a nearby meteorological station.

2. Data Pre-processing

When both data sets are obtained, it is essential to proceed with the preliminary screening phase to explore and understand better the data at hand. The data might contain some discrepancies such as missing values or outliers. Data cleaning must also be performed to fix the corresponding issues by removing outliers or dealing with the missing values- either filling the missing value or dropping the data. Therefore, both data sets are exported to Microsoft Excel and the following steps are performed:

- Check the features and the number of observations
- Check the time frame of the dataset
- Check the data type of each feature

- Check for any typing mistake
- Remove irrelevant features
- Handle missing data
- Fix any discrepancy
- Structure the datasets

After that, since time series is dependent on time, a time step for prediction (e.g., day, week, month, etc.) is specified depending on the available data and the forecasting tool's scope of work. The time factor is an important feature to consider in the production output prediction because activities are sequential in nature where time dependencies can exist between consecutive observations.

Consequently, it is essential to proceed with visualizing the output data to explore the patterns and understand better the data at hand. Also, the summary statistics of the output data should be obtained to further understand the corresponding data, and these include the mean, standard deviation, minimum and maximum value of the data.

A time series is simply a combination of signal (i.e.: trend, cycle, and seasonality) and noise. Once the target variable data is structured, an ETS (error, trend, seasonal) decomposition can be performed in order to visualize and investigate the trend, seasonal, and error patterns that exist within the data. It is helpful to split the time series into the various components where each represent an underlying pattern in order to explore and understand better the time series (Hyndman & Athanasopoulos, 2015).

At this stage, the screened and structured data must be pre-processed to fit the model as described by the following steps:

- **Feature Selection:** The features included in a model directly impact the performance and accuracy of this model. Several ways exist to select the suitable features, but the

most common approach is the correlation-based feature selection (CFS) because of its simplicity and effectiveness (Hall, 1998). Therefore, feature selection should be performed by studying the correlation (i.e., Pearson correlation or Spearman correlation) between each of the input variables (weather data and resource hours) and the output variable (production). Consequently, the factors that have significant correlation with the output variable will be selected (Hall,1998). To study the correlation, the Spearman correlation should be quantified between the input variables and the target variable and between the variables themselves. The Spearman correlation is a non-parametric correlation which measures the strength and direction of the monotonic relationship between two variables. The Spearman correlation can range between -1 and +1. A value of +1 indicates a positive relationship between the 2 variables, while the value of -1 represents a negative relationship between the 2 variables. A value close to zero indicates no relationship at all. Once the Spearman correlation is analyzed, the input variables that have a significant correlation with the target variable will be selected, and the attributes that provide no useful information for the task will be dropped. Also, the input features that seem to be significantly correlated with each other can be dropped in order to reduce the model's computational cost and improve the model's performance.

- **Feature engineering:** Feature engineering serves to improve the model's performance by suitably modifying the features. Useful features must be extracted so that the algorithm can easily interpret them (Petnehazi, 2019). The use of several features might cause noise on the model, so features could also be aggregated into promising new features.

3. ANN Data Preparation

a. ANN Input Identification

The ML model will incorporate the production output time series as an endogenous factor, so the previous production output will be used as input with the following time series as exogenous factors:

- The relevant weather data
- Resource hours

b. Data Scaling

Feature scaling is one of the main data transformations required for ML algorithms. This step standardizes the features in a range in order to handle the highly varying magnitudes of the various features. In general, ML algorithms don not perform well when the input numerical features have very different scales (Géron, 2019). Scaling enables neural network to converge faster (Ezeldin & Sharara, 2006). Two common methods exist to transform all the attributes to have the same scale: min-max scaling and standardization. Min-max scaling, known as normalization, transforms the features to a scale ranging from 0 to 1. It is achieved by subtracting the minimum value from the feature and dividing by the maximum minus the minimum value (Géron, 2019). However, the standardization method does not restrict values to a specific range (Géron, 2019). It involves scaling the features by subtracting the mean value from the feature, and then it divides the value by the variance in order to have a unit variance distribution (Géron, 2019).

c. Data split

To train properly the model and evaluate reliably its performance, the dataset should be grouped into three categories: training set, validation set, and testing set (Géron, 2019). After selecting the training algorithm, the training data set is used to fit the model to determine the corresponding weights and bias. The training error is determined according to the chosen performance metric. The trained model is then utilized to predict the output in the validation set, and the validation error is computed according to the adopted performance metric. Multiple models would have been trained with various hyperparameters using the training set, and these models would have been evaluated using the validation set to select the best performing model. However, this evaluation becomes biased because the error was evaluated multiple times on the validation set, so this adapted the model and hyperparameters to produce the best model for that set (Géron, 2019). Therefore, a testing data set is used to provide a test error and evaluate the final model.

Usually, this grouping into 3 categories is made randomly for most ANN models. However, because the sequence of data is important in a time series, the grouping must be performed in a sequential manner in order not to alter the sequence of time periods. It is important to adequately choose the training set size to have a well performing model. Also, the validation and test set sizes are typically chosen to be 10% of the data; however, in time series forecasting, the evaluation set sizes (i.e.: validation and test sets) can be chosen according to the maximum forecast horizon required (Shmueli & Lichtendahl, 2016). The forecast horizon is the length of time into the future for which forecasts are to be prepared. In other words, if the ultimate objective of the model is to know the predictions of the upcoming week or the next 7 days, the validation and test size can be

comprised of 7 data points. Forecast horizons can vary between long-term and short-term, and this is based on the scope of the model. This research focuses on improving the forecast accuracy during the construction of the project, so a short-term approach is adopted. In the literature, for instance Hwang (2010) used a 4 step-ahead forecast horizon for short-term planning in order to test the behavior of the developed model. So, in construction projects, the forecast horizon can differ from one case to another depending on how much the user needs to forecast in the short term to plan accordingly the resources. Intuitively, a range of a one week forecast horizon can be satisfactory to achieve reasonable planning; however, this depends on the application of the tool and the scope of the project.

4. ANN Model Development

An ANN time series predictive model is built to forecast field production based on the weather features, the resource hours used, and the previous production values. Python is the adopted platform to analyze and develop the corresponding models.

a. Neural Network type and Architecture Selection

Various types of neural networks can forecast time series data. In this research, only 2 types of neural networks will be investigated: Multi-Layer Perceptron (MLP), and Recurrent Neural Network (RNN) which includes Long-Short Term Memory networks (LSTM) and Gated Recurrent Unit (GRU).

i. MLP

A multi-layer perceptron model, which is a class of ANN, is a typical feed-forward neural network. The MLP is composed of input layers, hidden layers, and output layers. Each layer contains nodes or neurons that are linked to the previous layer's neurons by connections having specific weights (Géron, 2019). Every layer, except the output layer, contains a bias node that is connected to the consecutive layer. Furthermore, each node includes an activation function for mapping the neuron's summed input to its output. Figure 3 shows the architecture of a regular MLP.

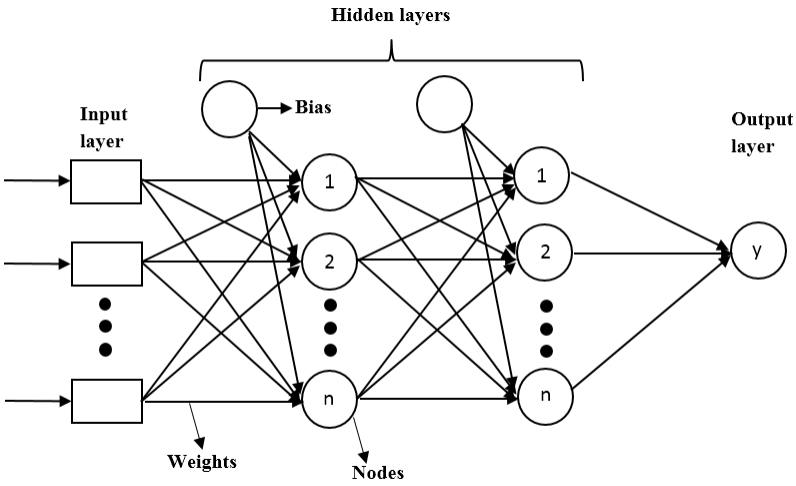


Figure 3 MLP architecture

Originally, the MLP network does not consider the time sequence of data and assumes that the input as well as the output data is independent of each other. However, time series forecasting can be established through MLP by taking a time lag or a window length and transforming the previous outputs as input factors to predict the target variable (Bianchi et al., 2017). The MLP architecture that is adopted is shown in Figure 4 ,where the previous outputs (y1) are used as input factors for the consecutive sample. Figure 5 also shows the MLP architecture which has the previous lags included as input variables.

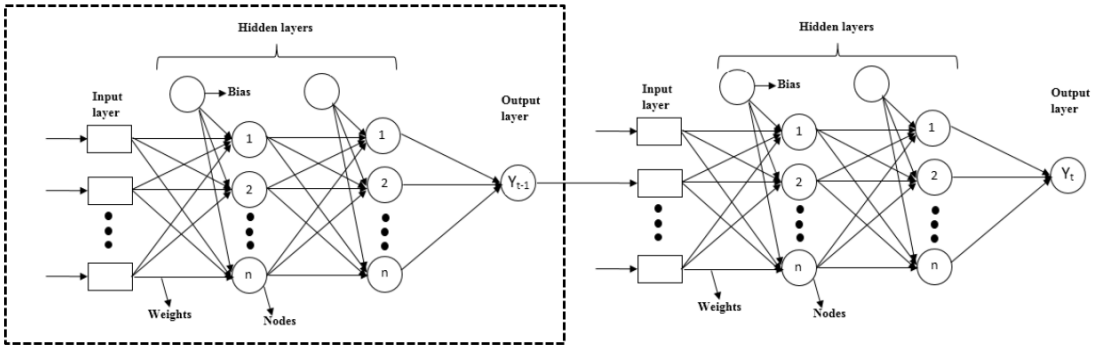


Figure 4 MLP time series architecture concept

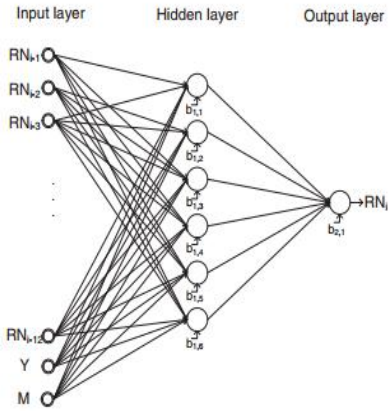


Figure 5 MLP time series architecture with previous lags

Even though this approach has been effective in several applications (Plumer, 2000; Díaz-Robles et al.,2008; Teixeira & Fernanades, 2012; Claveria et al., 2015), it only performs well if the temporal dependencies do not exceed the window length. Furthermore, it does not consider the temporal ordering as an explicit feature of the time series (Bianchi et al., 2017).

Therefore, a Recurrent Neural Network (RNN) can be more reliable in time series forecasting because it captures the temporal context within its feedback connections. As a result, they can capture the time varying dynamics of the underlying system (Schäfer & Zimmerman, 2007; Bianchi et al., 2017).

ii. RNN

A Recurrent Neural Network (RNN) can be adopted due to its substantial predictive ability which was demonstrated to outperform statistical time series methods utilized for the same objective (Laptev et al.,2017; Bianchi et al., 2017; Yunpeng et al.,2017). An RNN is a class of artificial neural networks (ANN), and it is specialized in the processing of sequential data or time series data to predict an output (Géron, 2019). The connections of an RNN look similar to the typical feed forward neural network but with internal connection nodes, or feedback loops, pointing backward (Géron, 2019). Although traditional neural networks assume that inputs and outputs are independent of each other, the output in a recurrent neural network depends on the prior elements within the sequence. Hence, the past values of the predicted target variable are used as a predictor to forecast future values of the target variable. Figure 6 shows the RNN architecture where the internal connections between consecutive outputs are observed.

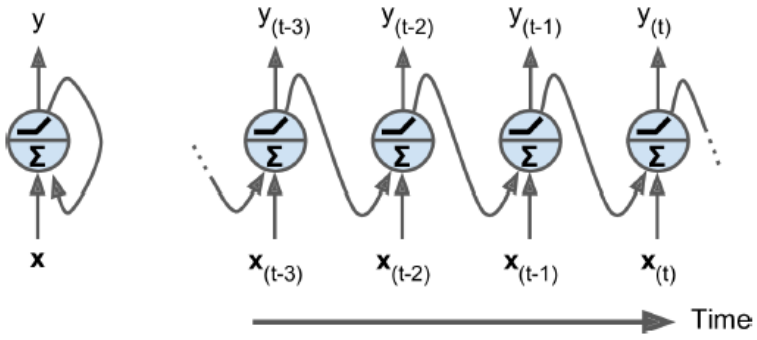


Figure 6 RNN architecture

RNN exhibits a dynamic behavior because of the time dependency of the data by allowing previous outputs to be used as inputs while having hidden states. The RNN has the ability to conduct a non-linear mapping from the input to the output when the relationship between the data is unknown (Bianchi et al., 2017). In addition to using previous outputs

to be used as inputs, the RNN model can also take additional exogenous factors for the corresponding time period as input.

RNNs are distinguished by their "memory" since they process information from prior inputs to influence the current input and output (Géron, 2019). However, RNNs have a major drawback in their memory capacity since they tend to gradually lose memory along the training procedure (Géron, 2019). Consequently, several RNN architectures with long–term memory, such as Long-Short-Term-Memory (LSTM) and Gated Recurrent Unit (GRU), were developed to improve the memory capacity of the network (Géron, 2019).

iii. LSTM architecture

The Long Short-Term Memory (LSTM) is a type of RNN. It is commonly used due to its superior performance in accurately modeling both short- and long-term dependencies in data (Géron, 2019). Long short-term memory is a gated memory unit for neural networks developed by Hochreiter and Schmidhuber (1997). It includes 3 gates (forget, input, and output gate) that manage the contents of the memory (Géron, 2019). The corresponding gates are logistic functions of weighted sums that are determined through backpropagation. The LSTM is distinguished by learning the required insights as well as remembering and recalling what is needed along the process. The forget gate and the input gate manage the cell state, that is the long-term memory. Also, the output gate produces the output vector that is the memory to be used. As a result, the following memory system enables the network to remember for a long time.

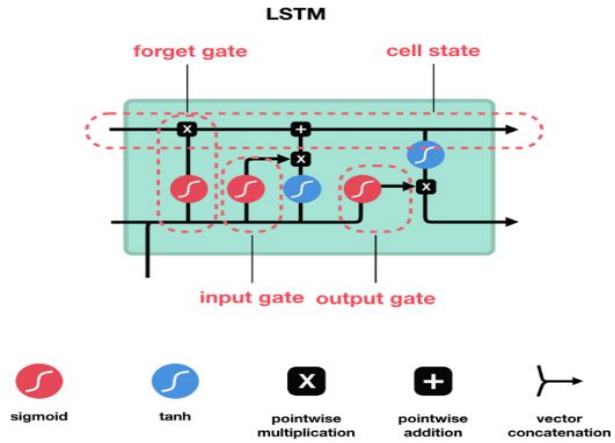


Figure 7 LSTM architecture

iv. GRU

Gated recurrent (GRU) unit also belongs to RNN category. GRU is a simplified LSTM developed by Cho et al. (2014). It has the same role in the network as the LSTM and seems to perform just as well as the LSTM (Greff et al., 2016). The major difference between the two is in the number of gates and weights. The GRU has only 2 gates. It lacks control over the memory content because it does not have an output gate. A single gate controls both the forget gate and the input gate. The update gate is responsible to add new information and to control the information flow from the previous activation, and the reset gate is inserted into the candidate activation (Géron,2019). Overall, the GRU is similar to LSTM. However, the differences between them makes it challenging to know which one is more suitable for a given problem.

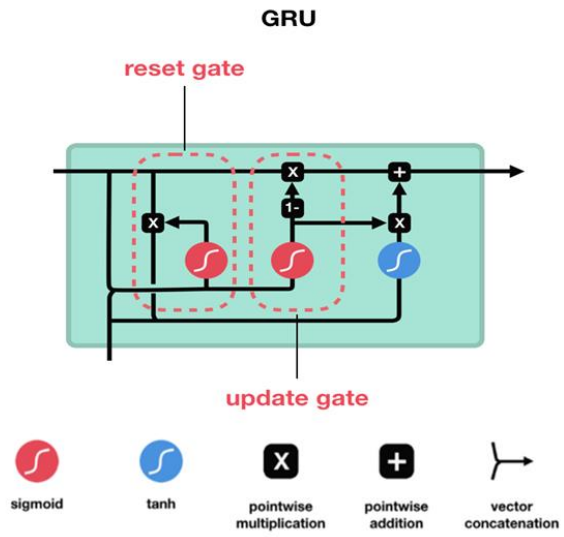


Figure 8 GRU architecture

b. Performance Function Selection

Several metrics are available to measure the performance of the model, such as mean square error (MSE), mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R-squared), etc. Although some similarities exist between each metric, each method represents the performance and punishes the error differently. A typical generally preferred performance measure for problems is the Root Mean Square Error (RMSE) (Géron, 2019). It measures the standard deviation of the errors the system makes in its predictions.

Equation 5, 6, and 7 represent the MSE, RMSE, and coefficient of determination (R^2), respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \qquad \text{Eq. (5)}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \qquad \text{Eq. (6)}$$

$$R^2 = 1 - \frac{sum\ squared\ regression\ (SSR)}{total\ sum\ of\ squares\ (SST)} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad \text{Eq. (7)}$$

Where n is the number of samples in the data, y_i is the actual data point value, $\hat{y}_i$ is the predicted data point value, and $\bar{y}$ is the mean of the actual values

c. Number of Time Lags Selections

The aim of the time series is to utilize and model the time dependencies within the data to capture the patterns over time. The time lags are the previous time steps of output values that will be used as input predictors in the forecasting process. For instance, consider an ANN model that has 4 input features and 1 target variable (i.e., output value) needed to be predicted. If a number of 2 time lags was selected, then the output target variable of time step 1 and time step 2 will be used as an input feature to predict the target variable of the upcoming time step which is time step 3. So, two input features will be added to the current 4 input features, thereby producing a total of 6 input features. Also, because a lag of 2 was chosen, the first 2 time steps will be utilized for initialization; as a result, the first prediction will start at the 3rd time step. Through this restructuring, the dataset is transformed into a time series forecasting problem. Therefore, a suitable time lag of production values that includes significant time dependency must be chosen for an adequate fit. The number of time lags can be specified based on the behavioral patterns present in the data, and it can be incrementally increased by beginning with a minimum lag of one until the model begins to underperform.

d. ANN Hyperparameters Selection

The hyperparameters of an ANN are variables that should be user specified prior to the learning process and are not affected by the learning algorithm itself. The MLP and RNN networks have the ability to learn from the continuous non-linear mapping of input to output data. The networks are comprised of layers having a number of nodes or neurons, and the neurons are connected to each other by means of the edges. The neurons connections have a weight that is adjusted as learning proceeds. Each neuron sums the value of the previous connected neurons after applying the corresponding weight for each element. Following that, an activation function is applied to the obtained sum to proceed with the mapping process. Various type of activation functions are present, and the common types are linear, sigmoid, hyperbolic tangent, and the rectified linear unit functions (RELU) (Nwankpa et al., 2018).

The learning process of the ANN model is performed through an optimization algorithm that aims to minimize the model error to determine the corresponding weights and bias. The weights and bias are optimized to produce output values that are approximate to the actual ones. Several training algorithms exist to solve non-linear optimization problems, and the most common ones are gradient descent, Gauss-Newton, and the Levenberg-Marquardt methods (Battiti, 1992; Gavin, 2011).

For every training sample, the algorithm inputs the values to the network, produces an output for every neuron in the layers, and computes the prediction (forward pass). Following that, the error value is measured by differencing the predicted and the actual values. Then, the error gradient or error contribution for the connection weights is computed through the backpropagation procedure. The backpropagation will efficiently compute the error gradient for each connection weight through reversing the error

gradient backward in the network (Géron, 2019). Finally, when the error gradient for each connection is computed, the connection weights are adjusted and updated through a specified optimization algorithm. Gradient descent is the most commonly adopted method where the error is minimized and fed back into the neural network to update weights accordingly (Géron, 2019). To mitigate the entrapment in a local optimum and improve the convergence of the model, several gradient descent optimization algorithms were developed, such as: Momentum, RMSProp, AdaGrad, Adam, and others (Géron, 2019).

The structure of the model in terms of the layers and nodes selection impacts the complexity of the model; therefore, they should be correctly picked to avoid both overfitting and underfitting. Underfitting happens when the model fails to sufficiently learn from the data and does not fit to the training data set. Therefore, both the train and test errors are considered to be high. To mitigate underfitting, it is essential to improve the complexity of the model by selecting a more powerful model by modifying the hyperparameters, checking the adequacy of the data, and improving the features used (Géron, 2019). Overfitting happens when the model exhibits a high complex behavior, so it behaves too well on the training set but cannot generalize to data beyond the training sample. As a result, a significant gap is produced between the train and test errors. To reduce the risk of overfitting problems, the model should be modified by either training the network on more examples, reducing the complexity of the network structure (changing the number of nodes or layers), or adding a regularization term (Géron, 2019). The regularization term penalizes the model by reducing its complexity and improves the generalization capabilities of the model (Bianchi et al.,2017), but it should be carefully selected to achieve an effective performance.

The main hyperparameters that need to be selected in order to build the ANN model include:

- **Number of hidden layers**

  It is recommended to begin by modeling two layers (one hidden layer, and an output layer); if the network performance with two layers is not powerful enough, three layers can be used (two hidden layers, and one output layer) (Demuth et al., 2009; Géron, 2019).

- **Number of neurons per hidden layer**

  The power of the network increases when increasing the number of neurons in the hidden layers; however, this requires further computation and is more probable to produce overfitting (Demuth et al. 2009). An appropriate network must be large enough to provide an adequate fit.

- **Activation function**

  The common activation functions include: linear, sigmoid, hyperbolic tangent, and the rectified linear unit functions (Nwankpa et al., 2018). The rectified linear unit functions RELU activation function performs well on many problems and seems to perform faster computations compared with the other activation functions (Géron, 2019).

- **Weight initialization**

  In training a neural network, the values of the weights and biases must be tuned to optimize the network performance. The network must be trained from several various

initial weight conditions to have a good performing model. The network performance can be optimized and verified by testing various initial conditions (Demuth et al. 2009). Neural networks can sometimes suffer from unstable gradients during the backpropagation procedure, so different layers may learn at different speeds (Géron, 2019). As a result, the weights and biases are initialized according to specific widely used techniques including the Glorot initialization or the He initialization. The Glorot initialization technique can speed up the training process and resolves the issue of the unstable gradients (Glorot & Bengio 2010; Géron, 2019); in addition, the He initialization technique is effective to be used for the ReLU activation function (Géron, 2019).

- **Learning rate**

Depending on the chosen learning rate value, training the model can be slow if the learning rate is a small value; however, if the learning rate is high, the model may not converge to a minimum (Géron, 2019). Thus, a suitable learning rate must be chosen to provide a better performing model.

- **Optimizer (Momentum, RMSProp, AdaGrad, Adam,etc)**

In an ANN model, a training algorithm performs the learning process to identify the weights and bias of the model with the aim of minimizing the model error (i.e.: cost function). The weights and bias are optimized by enabling the model to give an output value which is close to the actual value. To mitigate the entrapment in a local optimum and improve the convergence of the model, several gradient descent

optimization algorithms were developed, such as: Momentum, RMSProp, AdaGrad, Adam, and others (Géron, 2019). The Adam optimization algorithm combined several concepts from other optimization algorithms, and it is recently proven to be the most effective by improving convergence and producing better performance (Kingman & Ba, 2015; Géron, 2019). Adam is a form of stochastic gradient descent (a variation of gradient descent) that iteratively updates network weights by using an exponential moving average of the gradient and the squared of the error gradient (Kingman & Ba, 2015). This optimization algorithm is suitable for non-stationary objectives and data with very noisy sparse gradient (Kingman and Ba, 2015). For the Adam optimizer, the "momentum decay hyperparameter $\beta 1$" is a hyperparameter that should be selected. $\beta 1$ ranges from 0 to 1. The value of 0 refers to high friction, while the value of 1 refers to the lack of friction. Usually, a value of 0.9 is utilized. In addition, the "scaling decay hyperparameter $\beta 2$" is another hyperparameter that should be selected, and it also ranges from 0 to 1. Often, $\beta 2$ is specified to be 0.999. Usually, those mentioned default parameters produce an adequate performance (Géron, 2019).

- **Number of epochs**

An increased number of epochs may lead to overfitting problems where the model is unable to generalize to unforeseen data. Therefore, the number of epochs of the training set passing through the algorithm must be adequate to achieve a good performing model. One epoch refers to one complete pass over all the data points in the training set. To determine the optimal number of epochs required, the loss function of the training set and validation set should be monitored during the training process.

When the loss functions become stable and stop decreasing, the number of epochs required is reached. Otherwise, beyond that point, the model starts to overfit.

- **Regularization weight**

As previously explained, a neural network is prone to overfitting due to the high number of hyperparameters (Géron, 2019). In the model, large model weights should be prevented because they show high complexity and will most likely fail to generalize well on unforeseen data. Regularization can be adopted to prevent overfitting by decreasing its complexity and calibrating the machine learning model to minimize the adjusted loss function (Géron, 2019). Regularization is controlled by a hyperparameter which should be user-specified in the training process. Ridge (L1) and Lasso (L2) are two of the common regularization methods that could be adopted. Both methods add a regularization term to the cost function which is being minimized by the training algorithm. The terms range from 0 to 1. Ridge regularization allows the learning algorithm to produce an adequate model fit while minimizing the weights as low as possible. Whereas the Lasso regularization method removes the insignificant feature weights during the training process, thereby automatically conducting a kind of feature selection. The method along with their corresponding hyperparameters must be selected carefully; in addition, a combination of both regularization terms could be utilized (Géron, 2019). In an ANN model, the weights are computed for each layer, but in practice, the regularization hyperparameter is selected the same for all layers to avoid tuning several hyperparameters (Goodfellow et al., 2016).

Several scenarios are developed based on the various time lag, ANN architectures, and ANN hyperparameters. The corresponding scenarios were executed various times in a proper way to ensure that the model is performing appropriately. Various methods exist in order to choose the better performing model with the optimum values of hyperparameters. The basic method is the trial-and-error method through iterations. Also, a grid search approach could be adopted to achieve a faster result. It relies on automating the trial-and-error process to select the best parameters from a list of selected parameter options.

When the hyperparameters are selected, the ANN model is fitted on the training data set. It is essential to monitor the performance of the training set compared with the validation and test set to ensure selecting a model that has a low error and has no overfitting issues.

Once the model is fitted, the model's train set error metrics are assessed. The model is considered to be performing well on the train set if it has a relatively low error given the context of the problem, and it should have an adequate model fit (R^2). Otherwise, the steps in this section should be repeated by modifying accordingly the hyperparameters or the number of time lags, and the model is re-fitted.

If the corresponding model shows an adequate performance on the train set, the model should be further evaluated on the validation set to ensure that the model with the selected hyperparameters is not overfitting on new unforeseen data. The walk-forward validation approach could be adopted. The walk-forward validation method is a method for time series models to cross-validate the ANN model. It involves walking through the data sequentially in the validation set by performing either one-step ahead forecasts or

multi-step ahead forecasts, and then evaluating the performance of the forecasts by computing the error metrics on the validation set.

Several ways exist for the walk forward validation forecast (Hyndman, 2014; Hyndman & Athanasopoulos, 2015):

- One-step forecasts without re-estimation. The model fits to a single set of training data, and then one-step ahead forecasts are performed on the remaining data sets.
- One-step forecasts with re-estimation. The model is re-fitted at each iteration before each forecast is performed. At each iteration, the model train set is expanded to include the next actual time step value.
- Multi-step forecasts without re-estimation. Similar to one-step forecasts, but its performed for the next multiple steps.
- Multi-step forecasts with re-estimation. An approach where the model is re-fitted at each iteration before each forecast is performed.

Either the one-step ahead forecast method or the multi-step ahead forecast method can be used to forecast values in a time series. One-step ahead forecast is the task of predicting one step prediction in time without incorporating the previous predicted time step value. However, the multi-step ahead forecast is the task of predicting a sequence of values in a time series, so it uses the previous predicted value of a time step to predict the upcoming value in the next time step. Both are feasible methods, but choosing the suitable method depends on the scope of the project and the application purpose of the tool. For resource planning purposes in a construction project, the multi-step ahead forecast method is the more suitable approach since it can allow resource planning along a longer time horizon.

If the errors seem to be low in validation sets, the framework steps proceed, otherwise the model is re-fitted by changing the number of lags or the selected hyperparameters. Once the train and validation error are checked to be adequate, the model with the selected hyperparameters should be checked for stability. Due to the stochastic nature of the ANN models, a stability check should be performed by fitting the model 30 times using the same hyperparameters. Through every iteration, the initial weights is the only parameter that is changing because of the stochastic nature of the neural network optimization algorithm. Both the train and validation set error metrics of the 30 runs were assessed by computing the average error and the standard deviation across the 30 runs.

If both the train and validation average error of the 30 runs are close to the selected model's train and validation error metrics, and the standard deviation is relatively low, the model is ensured to be stable. Otherwise, the steps in this section should be repeated by modifying accordingly the hyperparameters or the number of time lags, and the model is re-fitted. Following that, the model will be assessed on the test set by checking that the test error metric is low and close to the train and validation error. Otherwise, the steps in this section should be repeated by modifying accordingly the hyperparameters or the number of time lags, and the model is re-fitted.

Once the best fit model is produced, the obtained network for the predictive production model is saved with the corresponding weights and biases. Finally, the network is used to forecast the production output in the upcoming time steps where the user must input the exogenous factors related to each time period.

5. Forecasting Application Implementation

Once the model is properly trained and tested, the developed predictive tool could be used by the PM to accurately forecast the production for the upcoming days while accounting for the available equipment resource hours of the upcoming week, the weather conditions of the upcoming week, and the production of the past few days. This allows the contractor to have more accurate estimates of the production based on the given conditions, and it reduces the uncertainty in anticipating the fluctuating productivity. In addition, it allows for a more accurate planning of the equipment resources since the PM can know, given these conditions, the production output of the upcoming days. This way, data driven decisions can be made to ensure that the equipment could be made available at the right time along with the required resource hours in order to meet the schedule and avoid delays.

In terms of frequency of use, the model may be run on a weekly basis with the results feeding into a short-term lookahead schedule. Beyond a 7-day window, weather forecasts become less reliable, and the resource status availability becomes tentative.

a. <u>Input identification</u>

As a first step in the forecasting procedure, the PM should know the corresponding **input** features of the model as follows:

1. **The weather data** for the *next week* should be obtained from the weather forecasts. A 7-day weather forecast is usually 90% accurate, so these input estimates will be close to what will really happen in the next 7 days.
2. **The resource hours available** for *next week* according to what resources are available and what is the PM planning to use for each day should be

specified. Based on the PM's experience, the available resources, and the schedule requirements, a baseline resource plan is established for next week by specifying the resource hours to be used for each day. This resource plan would be the first iteration in the model when utilizing the forecasting model. Based on the forecasted production output, these values can be adjusted in the "resource plan adjustment" step in order to meet desired production values.

3. **Production of the *previous days are*** obtained from the collected field production data.

b. Model Output

The PM inputs the features into the developed model. Consequently, the model forecasts as an output the production (quantity/day) for the next week for each day.

c. Resource Plan Adjustment

At this stage, the PM checks if the production output forecast is satisfactory. If so, the PM can adjust the baseline resource hours to proactively plan resources ahead of time in case there is a problem in production output at any specific time period. After adjusting the hours, the PM can run another forecast trial of the model to check again the production output forecasts. So, through iterations, the PM adjusts the resource management plan based on the production forecast values and gets new forecast values according to the specified resource hours. The resource plan is then specified, and the PM gets the forecast of production output for each upcoming day.

For example, if the activity being performed is excavation work in a project. Given the weather conditions of next week, the baseline resources planned to be utilized for the next week, and the recent excavation experience, the PM finds that the production output that could be achieved is much lower than that from the desired amount as per schedule requirements. For example, Figure 9 shows a sample of this scenario.

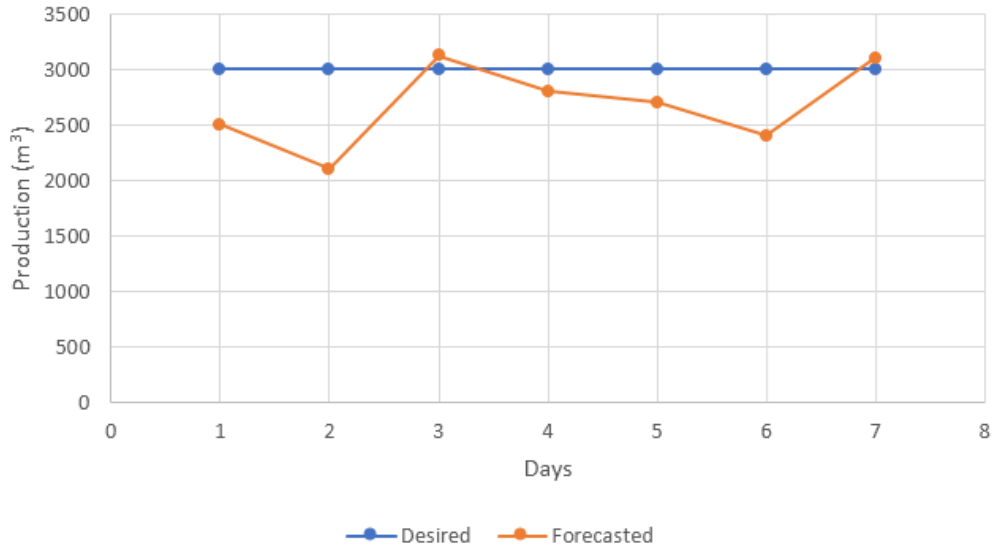


Figure 9 Example of forecast output

One solution could be that the PM could adjust the resource hours iteratively until the forecasted production output of the model is close to that desired production. This way, the resources are utilized and managed in a systematic way to prevent any schedule overrun and to meet the desired production values as per schedule requirements.

Another scenario would be if the PM finds that the production output is much higher than that from the desired value. Consequently, the PM adjusts the resource hours iteratively until the forecasted production output is close to the that desired production value. These additional resource hours could be used on other tasks or even other projects, otherwise they would have been idle during this task.

6. Update the Model

After the forecasting process occurs, a resource management plan is planned and executed. Consequently, the actual field production and weather parameters are tracked on site. The new collected data are then logged in the model, and the model should be updated for it to learn from the new actual data.

B. Benchmark Models

1. Statistical Time Series

It is important to have a benchmark model to be compared with the ML time series model performance. Therefore, powerful statistical time series models, that incorporate the same input features as that of the ML time series model, are developed and presented in this section for comparative purposes with the proposed ML time series model.

As previously mentioned, the patterns that may be detected in a time series are trends, cycles, seasonality, and residuals (Hyndman & Athanasopoulos, 2015).
Various statistical methods are present for modeling and forecasting time series data, such as moving average (MA), exponential smoothing, auto-regressive integrated moving average (ARIMA), and others. The ARIMA model, which is developed by Box and Jenkins (1976), is a classical and powerful time series model widely used and proven to be useful in many given problems (Hyndman & Athanasopoulos, 2015). It is able to predict the future value of a variable through a linear function of several past observations and random errors. The ARIMA model of a time series is defined by three terms (p, d, q) (Tabachnick & Fidell, 2018). The model can also account for time series that exhibit seasonality through the seasonal ARIMA (SARIMA) model. In addition, the model can also account for exogenous factors (external input features) by using the ARIMAX or

SARIMAX models- where the “X” stands for exogenous. The performance of the following statistical time series models can be developed and investigated:

1- ARIMA

2- ARIMAX; having the same exogenous factors of the ones used for ANN

3- SARIMA

4- SARIMAX; having the same exogenous factors of the ones used for ANN

The time series analysis methodology has the following standard sequential procedure: investigate the main features of the time series, check dependency among datapoints, pick an adequate model to fit the series, diagnose and evaluate the constructed model, and forecast and update (Brockwell & Davis, 2002). The following subsections present the steps for developing the statistical time series model.

a. Data Collection and Preparation

This section deals with collecting the data as the steps used in the generic framework as per section one Data Collection. The data collected includes the field production data and weather data.

The data is prepared and pre-processed as per section two Data Pre-processing of the generic framework.

The data is split into 3 sets: training, validation, and test sets. The training set is used to develop and train the model. The best performing models is chosen according to the validation set result errors. Then, the chosen models is evaluated on the test set to pick the final model. The split is performed in a sequential manner to respect the time sequence in the time series model.

b. Stationary Time Series

The first step of investigating the main features of the time series involves plotting the time series data and detecting the presence of a trend, a seasonal component, any significant change in behavior, or outlying observations (Brockwell & Davis 2002). The data must be stationary for it to be properly used in the ARIMA model. The classical ARIMA models perform well with stationary time series. As a result, it is crucial to perform transformations on the data series to make it stationary for it to be used in the ARIMA model (Hyndman & Athanasopoulos, 2015). A time series is said to be stationary if both its mean and its variance (amplitude) remain constant through time. Time series, that have trends or seasonality, are not stationary because the trend and seasonality affects the value of the time series at different times.

The time series stationarity can be tested either by visualizing the plotted time series, by applying statistical tests, such as the augmented dickey fuller (ADF) test, or by plotting and investigating the auto-correlation function (ACF) plots of the time series (Hyndman & Athanasopoulos, 2015).

As a first step, the ADF test can be applied. The ADF test is a common statistical test used to check whether a time series is stationary or not. The null hypothesis of the ADF test is that the time series is non-stationary. So, if the p-value of the test is less than the selected significance level (i.e.: 0.01 or 0.05), the null hypothesis is rejected and the time series is inferred to be stationary. Otherwise, the series is considered to be non-stationary; thus, in this case, the series must be transformed to achieve a stationary state. Also, the stationarity of the series can also be detected by visualizing the ACF plots of the series. Similar to the statistical correlation that represents the strengths of the relationship between two variables, the ACF measures

the linear relationship between the time series and *lags* itself (Hyndman & Athanasopoulos, 2015). A series is non-stationary if the ACF plot exhibits a behavior of gradual decay to zero or has positive autocorrelations out to a high number of lags (Nau, 2016). In a stationary time series, the ACF will drop quickly to zero quickly.

Several ways exist to make a time series stationary either by removing the trend or the seasonality. The most adopted method is the differencing method (Hyndman & Athanasopoulos, 2015). The differencing method computes the difference between consecutive observations to remove the trend, or it can compute the difference between consecutive seasons to remove the seasonality. Differencing eliminates or reduces trend and seasonality, thereby helping to stabilize the mean of a time series by removing changes in the level of a time series (Hyndman & Athanasopoulos, 2015).

The right value of differencing is the lowest order of differencing which produces a time series that "fluctuates around a well-defined mean value and whose autocorrelation function (ACF) plot decays rapidly to zero, either from above or below" (Nau, 2016). The non-seasonal integration order "*d*" parameter is defined according to the specified non-seasonal differencing value to make the series stationary. In case seasonality is present, the seasonal integration order "*D*" parameter is also defined according to the specified seasonal differencing. Over-differencing should be prevented; hence, it is suggested not to use more than one order of seasonal differencing or more than 2 orders of total differencing (seasonal and nonseasonal) (Nau, 2016).

c. ARIMA (*p*,*d*,*q*) (*P*,*D*,*Q*) *m* development

i. Identification

The ARIMA model has three parameters that need to be identified: p, d, and q.

- ***p* = order of the autoregressive part**

  The "AR" term in ARIMA refers to the *auto*regression, which is represented by the "*p*" term. The AR term forecasts the variable of interest using a combination of *past (or lagged) values of the variable*. Thus, this indicates that it is a regression of the variable against itself (Hyndman & Athanasopoulos, 2015). The p term represents the number of lagged values used for the forecast.

- ***d* = degree of first differencing involved**

  The "I" term stands for Integrated, referring to the differencing "*d*" value needed to achieve stationarity as mentioned in the previous section "Stationary Time Series".

- ***q* = order of the moving average part**

  The "MA" term refers to the moving average, which is represented by the "*q*" value. The moving average uses past forecast errors instead of using past values of the forecast variable in a regression (Hyndman & Athanasopoulos, 2015).

The ACF and partial auto-correlation function (PACF) plots are significant plots that indicate the corresponding signals, patterns, and relationships in the time series data (Tabachnick & Fidell,2018). Therefore, the *p, d,* and *q* terms can be identified according to the ACF and PACF plots. So, as a first step, the model initial parameters should be estimated based on the basic principles of the ARIMA model order selection by relying on the ACF and PACF plots.

Time series analysis is effective and meaningful only when dependencies are present among the observations. After the time series has become stationary, the following step is to check the correlation dependency within the time series itself by

plotting the ACF and PACF plots of the differenced series. The ARIMA orders are identified through patterns in their ACF and PACF; hence, it is essential to plot both graphs and determine the underlying patterns in each one (Tabachnick & Fidell,2018). These plots summarize graphically the strength of the relationship for an observation in a time series with observations at prior time steps.

Similar to the statistical correlation that represents the strengths of the relationship between two variables, the ACF measures the linear relationship between the time series and *lags* itself (Hyndman & Athanasopoulos, 2015). The ACF can be used to determine an estimate of the MA order "*q*" parameter of the model (Nau, 2016). The autocorrelation between one observation and another observation at a prior time step includes the direct and indirect correlations.

The PACF are self-correlations with the intervening autocorrelations removed. The PACF remove the indirect correlations that are computed by the ACF. The PACF can also be used to estimate the AR order "*p*" parameter of the model (Nau, 2016). Therefore, the PACF describes the direct relationship between an observation and its lag. The "*p*" and "*q*" parameters are chosen according to the number of significant correlations present for each of the PACF and ACF plots, respectively. The basic principles of the ARIMA model order selection are summarized in Table 3. . Considering an AR process only, the ACF plot gradually decreases with the PACF performing a sharp drop after "p" significant lags. For a MA process, the ACF shows a drop after a certain "q" number of lags, and the PACF should demonstrate a gradual decreasing shape. Finally, if both ACF and PACF plots show a gradual decreasing pattern, then the ARIMA process is considered for modeling.

Table 3 Principles of the ARIMA model order selection

	AR(p)	MA(q)	ARIMA(p,q)
ACF	Tails off	Cuts off after q	Tails off
PACF	Cuts off after p	Tails off	Tails off

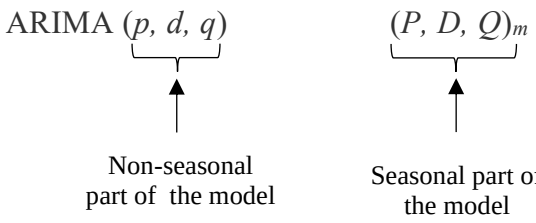
For the integration parameter "*d*", the order is selected based on the differencing order selected which was explained in the previous section. The equation of the ARIMA model is the following:

$$\boldsymbol{y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots \beta_p y_{t-p} + \in t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots \theta_q \epsilon_{t-q}}$$

Eq. (4)

where, y_t denotes the value of the time series at time t, $\beta_1 y_{t-1} + \beta_2 y_{t-2} + \ldots + \beta_p y_{t-p}$ denotes the autoregressive part which includes the previous p lagged values of the time series, and $\epsilon t + \Theta_1 \epsilon_{t-1} + \Theta_2 \epsilon_{t-2} + \ldots + \Theta_q \epsilon_{t-q}$ denotes the moving average part that includes the error terms at time t and past times till $t - q$. As previously mentioned, p and q represent the autoregression and the moving average orders respectively. β_1 to β_p represent the autoregression (AR) coefficients, and θ_1 to θ_q represent the moving average (MA) coefficients.

The time series can also vary periodically by exhibiting a seasonal behavior. ARIMA models are also able to model a wide range of seasonal data. In case seasonality is present in the data, the seasonal ARIMA (SARIMA) is developed by including additional seasonal terms (*P, D, Q* and *m*) in the ARIMA model. Since the data might include seasonal patterns, the SARIMA is an option to consider.



Although the seasonal *P*, *D*, and *Q* components have similar definition of the non-seasonal components, they involve back-shifts of the seasonal period. The seasonality can be identified in the ACF and PACF plots. When the data has a seasonal behavior, the autocorrelations will produce spikes for the seasonal lags at multiples of the seasonal frequency. The seasonal lags of the PACF and ACF plots will indicate the seasonal part of an AR or MA model (Hyndman & Athanasopoulos, 2015). The "*P*" and "*Q*" parameters are chosen according to the number of significant seasonal lags present for each of the PACF and ACF plots, respectively. The *m* term represents the seasonal period that is obtained through visualizing the seasonal pattern in the time series plot or by investigating the ACF plots where spikes appear at the seasonal lag. The SARIMA order determination step follows the same procedure previously described for determining the order of the ARIMA models.

When exogenous or external factors are to be included in the model, the ARIMAX or SARIMAX model can used. For the ARIMAX or SARIMAX models, the exogenous factors will include the same input of weather and resource hours used in the ANN model in order to undergo the comparison between both.

ii. Estimation

Once the orders are identified, the software proceeds to compute the coefficients or parameters of the model equation through the Maximum Likelihood Estimation (MLE)

method (Hyndman & Athanasopoulos, 2015). This method identifies the values of the coefficients that maximize the probability of obtaining the data that is observed.

The process of correctly determining the orders from the ACF and PACF plots is subject to various rules that must be followed accordingly (Hyndman & Athanasopoulos, 2015). However, the order selection based on the ACF and PACF plots might be misleading and hard to interpret, so it remains essential to test the several combinations of orders to check their goodness of fit and corresponding performance (Nau, 2016). So, several iterations with different parameters will be performed to select the best performing model for the data at hand. Selecting the best performing model is based on evaluating the data set according to the error metrics adopted. The root mean squared error (RMSE) is usually the most preferred metric to be used (Géron, 2019). The MSE, RMSE and R^2 of the training and validation sets will be recorded and evaluated. A model performs well on the training set if it has a relatively low error metric according to the adopted performance function and according to the context of the problem. Also, the model's Akaike Information Criterion (AIC) should be considered in order to choose a simpler model with a better goodness of fit. The AIC is an important value to consider in the model selection since it evaluates how well a model fits the data model by considering a model's simplicity and goodness of fit (Hyndman & Athanasopoulos, 2015). The AIC penalizes complex models that use more independent variables in order to avoid overfitting problems; therefore, the model having the minimum value of the AIC is normally the best model for forecasting (Hyndman and Athanasopoulos, 2015). The AIC can be used to compare ARIMA models with different parameters in order to select a simpler model with a better fit.

A reliable guide to selecting proper terms in an ARIME model is to assess the model's performance metrics and choose the model with a lower AIC by applying validation techniques (Hyndman & Athanasopoulos, 2015).

At this stage, the performance of the model will be assessed based on the error metric values of the training sets. If the model is satisfactory on the training set, the walk-forward validation method will be performed on the validation set to further evaluate the model's forecasting performance.

iii. Diagnosis

In this step, the residuals of the model must be checked to verify that the model has captured all the information which can be modeled (Hyndman & Athanasopoulos, 2015). The residual plots should be produced and analyzed in order to check the goodness of fit of the model. The residual is the difference between the training sets' actual value and the model's training prediction values. To obtain a good model, the residual should not be correlated, and this can be determined by plotting and analyzing the ACF plot of the residuals. If the residuals are significantly correlated, then there is information that was not modeled. Consequently, the model parameters must be identified as is in step one. Also, the residuals should have a mean close to zero, a constant variance, and a normal distribution (Hyndman & Athanasopoulos, 2015). So, the histogram of the residuals should be plotted and analyzed to determine if these conditions apply (Hyndman & Athanasopoulos, 2015).

When an adequate model is selected with the appropriate residual diagnostics, it can be evaluated on the test sets in a walk-forward approach. If the corresponding results are not adequate, the model parameters can be updated, and

the steps are once again repeated. The obtained error is then compared with the MLP model error.

2. MLP model with no Time Lag

A similar methodology is applied to an MLP model that does not have the time factor included. The purpose of this model is to compare its performance with the proposed ML time series' performance to show the importance of including the time dependency in the data.

The model development is established by randomly shuffling the data points and not accounting for the time dependency in the data. The same methodology of the generic framework is followed, but the difference is that the data split is performed randomly rather than sequentially. Also, the number of lags is not included as a predictor, thereby the time dependency is not accounted for.

CHAPTER V

RESULTS AND DISCUSSION: CASE STUDY

The following chapter presents the detailed results and analysis of applying the proposed methodology and time series model on a case study. The case study focuses on data of excavation works of a real-world large infrastructure project. Section A entails the project description of the case study. Following that, section B evaluates the developed generic framework of the ANN time series model. Then, section C presents the results of the benchmark models by testing the benchmark statistical time series ARIMA model along with the MLP model with no time lag. Once the models are developed, a comparative study is performed in Section D to evaluate the performance of the ANN time series model compared with both the statistical time series model and the MLP model with no time lag. Section D also includes comparing the results of the developed tool with the strategy adopted on the project with the goal of quantifying the improvements in the forecasting process.

A. Project Description

The corresponding project is a large infrastructure project in the Middle East. The excavation activity constitutes a major part of this project since it spans approximately two years. This continuous activity tends to be dynamic in nature where the productivity of the major resources fluctuates throughout the execution of project. In this project, the project manager forecasts the production values of each upcoming week by relying on subjective judgement, historical data of previous performance, and schedule requirements. However, the project manager's forecasted production estimates are being

either overestimated or underestimated compared with actual production. As a result, the project manager faces uncertainty in estimating the production values of the resources for the upcoming week. Therefore, he is facing difficulty in managing the resources on site to meet the desired production values. Consequently, a significant gap between the actual production and the project manager's forecasted production estimates is being experienced. This gap can be explained by the dynamic behavior of project over time where of resources fluctuate due to internal operational factors and external environmental factors.

Since the start of excavation, the project manager was able to collect production data through daily report logs that include production output and resource input used. Also, weather data from the project's weather station is available, and it includes the weather parameters for each working day. Although the available data covers an entire year, the case study is tested assuming that the project had started and is currently somewhere in its execution phase timeframe. The timeframe should be selected in a way to have sufficient data points to train the model. In this case study, 8 months of data (or 245 data points) were used.

B. Application of the Developed ANN Time Series Generic Framework

1. Data Collection

a. Production Data

The daily site report logs were obtained from the contractor. These logs contain the resources used on each day and the daily production data. The following table is a snapshot of the raw data of the resources used for each day. Resources are categorized by labor and equipment groupings. The resource hours and their corresponding description

are found in the production data. For instance, the labor manpower includes foreman, skilled labor, unskilled labor, driver, and others. While for the equipment, some examples include the excavators, the articulated haulers, the backhoes, and several others. Also, the equipment brand, type, and equipment details can be found in the daily site report log of Table 4.

Table 4 The obtained Daily Site Report Log

DAILY SITE REPORT LOG

Location	Date	Manpower	Equipment	Brand	Type	Equip. Details	Flywheel Powe	Cap. (m^3)	Quantity	From	To	Total Hours
Location 3	8/6/18	FOREMAN							1	7	13	6
Location 3	8/6/18	DRIVERS							8	7	13	48
Location 3	8/6/18		EXCAVATOR	VOLVO	480DL	EXCAVATOR (50 T)			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A35	ARTICULATED HAULER A35			1	7	13	6
Location 3	8/6/18		ARTICULATED HAULER	VOLVO	A40	ARTICULATED HAULER A40			1	7	13	6
Location 2	8/6/18	DRIVERS							7	7	18	77
Location 2	8/6/18	DRIVERS							4	18	21	12

In addition, the daily production output can be found in the following data set. For each day, the actual executed amount of excavation is recorded. Also, the total planned amount that the project manager estimated prior to one week from the actual execution can be found in the daily production log of Table 5.

Table 5 The obtained Daily Production Log

DAILY PRODUCTION LOG

Date	L1 - Planned	L1 - Executed	L2 - Planned	L2 - Executed	L3 - Planned	L3 - Executed	Total Planned	Total Executed
8/6/18	1084 m³	3436 m³	806 m³	2724 m³	2509 m³	912 m³	4399 m³	7072 m³
8/7/18	1084 m³	1844 m³	806 m³	2120 m³	2509 m³	3000 m³	4399 m³	6964 m³
8/8/18	1084 m³	1264 m³	806 m³	3040 m³	2509 m³	3000 m³	4399 m³	7304 m³
8/9/18	1084 m³	1334 m³	806 m³	920 m³	2509 m³	2160 m³	4399 m³	4414 m³
8/10/18	1084 m³	1100 m³	806 m³	2631 m³	2509 m³	1140 m³	4399 m³	4871 m³
8/11/18	1084 m³	2397 m³	806 m³	1170 m³	2509 m³	840 m³	4399 m³	4407 m³
8/12/18	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³
8/13/18	1084 m³	2224 m³	806 m³	1630 m³	2509 m³	2040 m³	4399 m³	5894 m³
8/14/18	1084 m³	516 m³	806 m³	3194 m³	2509 m³	1368 m³	4399 m³	5078 m³
8/15/18	0 m³	5820 m³	0 m³	4400 m³	0 m³	405 m³	0 m³	10625 m³
8/16/18	1084 m³	5228 m³	806 m³	2610 m³	2509 m³	0 m³	4399 m³	7838 m³
8/17/18	1084 m³	5760 m³	806 m³	2190 m³	2509 m³	0 m³	4399 m³	7950 m³
8/18/18	1084 m³	888 m³	806 m³	1892 m³	2509 m³	0 m³	4399 m³	2780 m³
8/19/18	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³
8/20/18	1084 m³	0 m³	806 m³	110 m³	2509 m³	0 m³	4399 m³	110 m³
8/21/18	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³
8/22/18	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³	0 m³
8/23/18	1084 m³	4956 m³	806 m³	170 m³	2509 m³	0 m³	4399 m³	5126 m³

b. Weather Data

The weather data was also recorded by the contractor's weather station located on site. The weather data contains the hourly weather data for each day. The weather parameters that can be found in this data set are the air Temperature (℃), dew point (℃), solar radiation (W/m^2), VPD pressure deficit (kPa), relative humidity (%), precipitation (mm), wind speed (km/h), instantaneous max wind speed (km/h), wind direction (deg), air pressure (kPa), solar panel (mv), and battery (mv). Table 6 shows the raw weather data obtained.

Table 6 The obtained weather data

WEATHER DATA

Log Details			HC Air Temperature [°C]			Dew Point [°C]		KZ Solar Radiation [W/m²]	VPD [kPa]		HC Relative Humidity [%]			Precipitation [mm]	Wind Speed [km/h		Instantaneous Max Wind Speed [km/h]	Wind Direction [deg]	Air Pressure [kPa]			Solar Panel [mV]	Battery [mV]
Date/Time	Date	Time	avg	max	min	avg	min	avg	avg	min	avg	max	min	sum	avg	max	max	last	avg	max	min	last	last
2018-08-06 08:00:00	8/6/18	8:00:00 AM	26.66	27.04	26.28	18.3	18.3	256	1.37	1.29	60.64	62.13	59.15	0	5.4	9.4	15.8	89	93.0081	93.014	93.0022	6676	6440
2018-08-06 09:00:00	8/6/18	9:00:00 AM	27.01	27.54	26.1	18.3	18	844	1.45	1.29	59.22	62	56.12	0.6	9	10.4	20.2	80	92.9915	93.019	92.9722	6753	6499
2018-08-06 10:00:00	8/6/18	10:00:00 AM	27.34	27.65	27.21	19.2	18.9	936	1.39	1.35	61.63	62.55	60.75	0	8.6	9.7	23.8	93	92.9659	92.976	92.949	6825	6556
2018-08-06 11:00:00	8/6/18	11:00:00 AM	27.08	27.56	26.24	19.5	19.3	740	1.29	1.09	63.66	67.95	61.25	0	9.7	10.8	25.2	94	92.9522	92.9582	92.9455	6849	6588
2018-08-06 12:00:00	8/6/18	12:00:00 PM	25.49	26.08	24.76	19.9	19.7	394	0.91	0.75	71.91	75.87	68.16	0	9	10.8	26.6	97	92.9418	92.9575	92.928	6844	6585
2018-08-06 13:00:00	8/6/18	1:00:00 PM	24.71	25.38	24.42	20	19.9	258	0.75	0.7	75.77	76.86	73.12	0	6.5	9.4	19.1	96	92.9315	92.9392	92.9257	6820	6573
2018-08-06 14:00:00	8/6/18	2:00:00 PM	24.05	24.67	23.73	20.1	20	147	0.61	0.56	79.26	80.68	76.14	0	5	5.8	13.7	89	92.919	92.93	92.9007	6810	6571
2018-08-06 17:00:00	8/6/18	5:00:00 PM	23.26	23.81	22.92	20.1	20	93	0.48	0.44	82.91	84.32	80.46	0	5.8	6.5	16.2	86	92.9062	92.9177	92.8962	6729	6548
2018-08-06 18:00:00	8/6/18	6:00:00 PM	22.8	23.01	22.58	19.9	19.8	52	0.43	0.39	84.11	85.75	82.74	0	5.4	6.5	13.3	94	92.9225	92.9312	92.917	2099	6516
2018-08-06 19:00:00	8/6/18	7:00:00 PM	22.02	22.5	21.67	20	20	3	0.29	0.23	88.81	91.02	86.25	0	3.6	6.1	11.9	94	92.9341	92.9432	92.9275	0	6494

2. Data Preprocessing

Both datasets obtained were consistent and clean without any missing values. As a first step, it is important to understand the data at hand. The data consists of 245 data points. Each day contains the corresponding production data and weather data.

After observing the datasets, a daily time step for prediction was selected since it is the step by which the data was available, the excavation activities were done, and the excavation equipment were planned. The daily time series for the excavation production data was developed. The time series plot of the actual production volume in cubic meter is shown in Figure 10. It is evident that the production of resources fluctuates over time along the execution of the project due to the dynamic nature of construction activities.

Figure 11 shows the time series of the estimated production volume value of the project manager. Due to subjective judgment and reliance on historical data, it is significant that the project manager assumes that the production is constant for every week interval and does not anticipate any change in productivity along the construction process.

Consequently, the actual production volume is plotted with the forecasted values of the project manager on the same graph in Figure 12. It is evident that there is a discrepancy between the actual volume and the project manager's forecasted volume. For

an interval period of one week, the actual production volume tends to fluctuate each day; however, the forecasted production values of the project manager tend to be constant along all the days of the week. Therefore, the planned values are either overestimated or underestimated at several time periods during execution. This would indicate that the project manager was put under uncertainty at various instances along the project.

Figure 13 shows the time series plot of the difference between the PM's forecasted production volume and actual production volume. Major differences exist where sometimes a difference of 10,000 m^3 is encountered. These spikes show the major discrepancy between what really happened and what was planned to happen by the PM. The positive values above the zero axis indicate instances where the planned values were overestimated by the contractor; as a result, the actual values turned out to be much less than what was planned to happen. On the other hand, the negative values show the instances where the contractor underestimated the planned values.

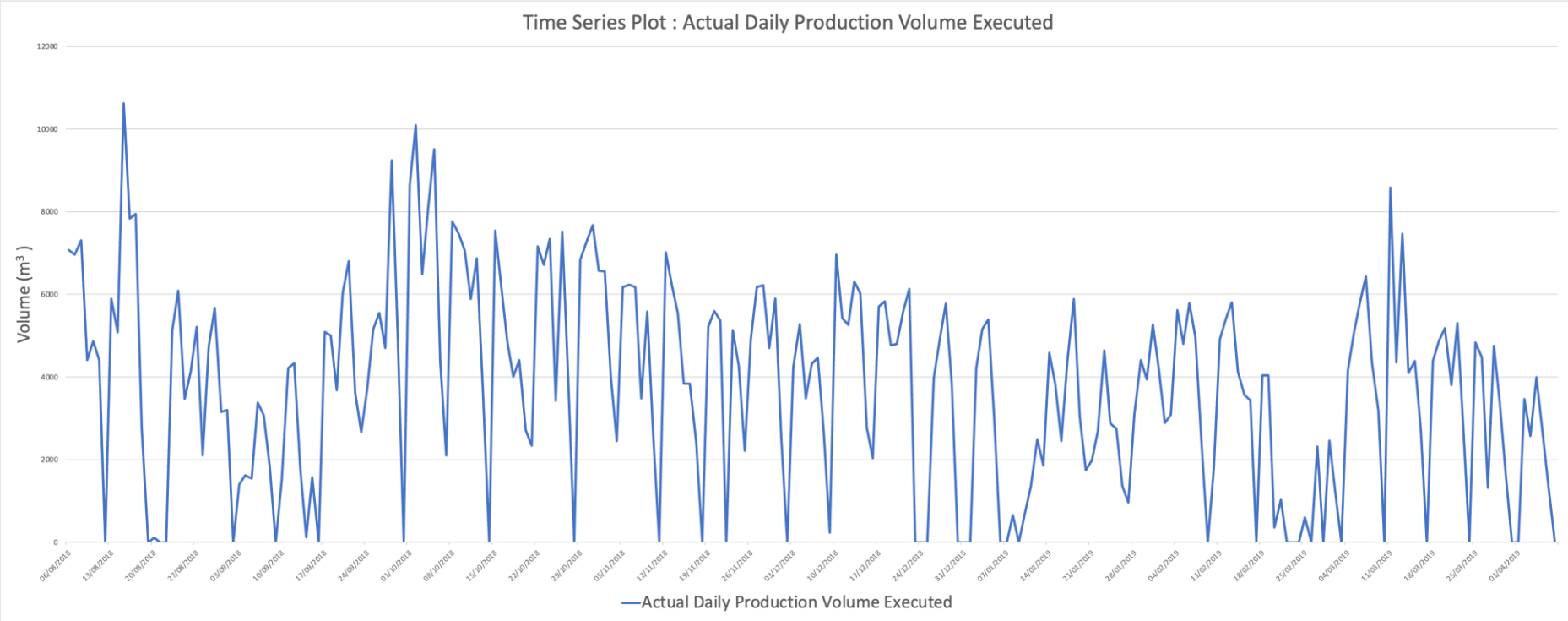


Figure 10 Time series plot of the actual daily production volume executed

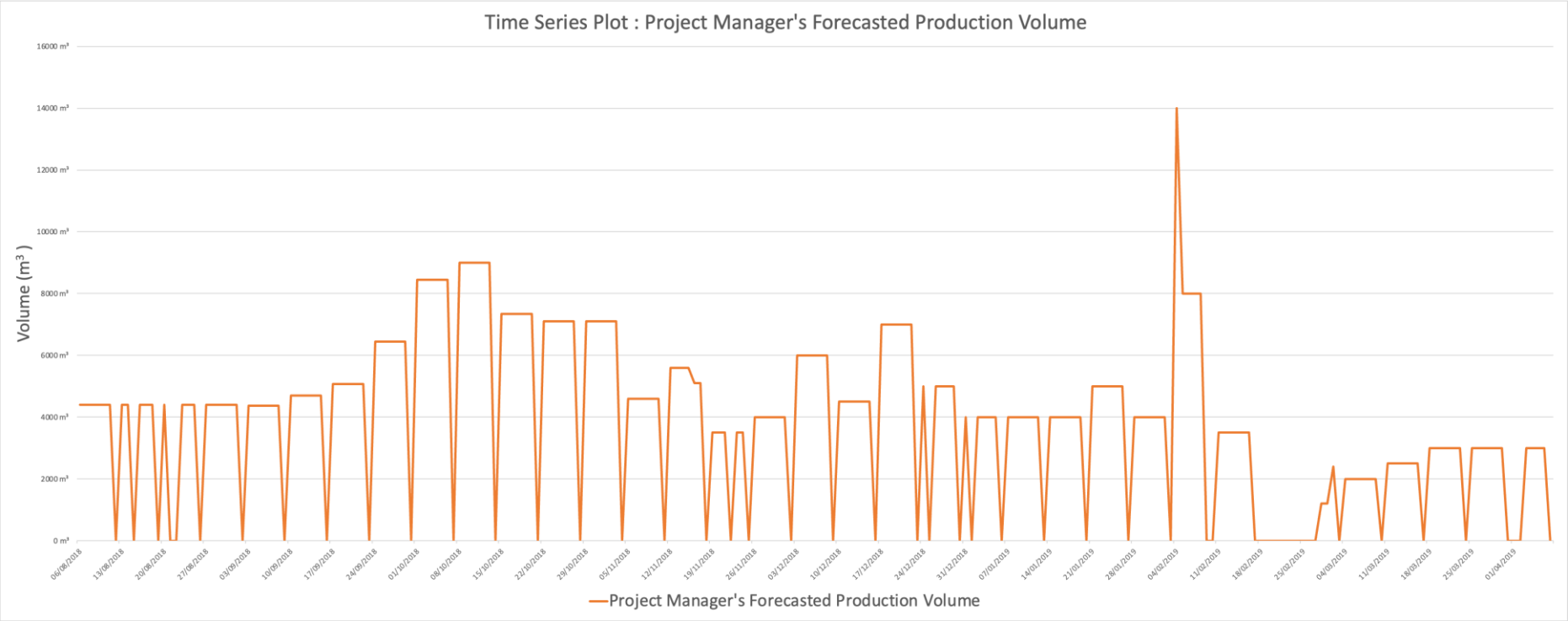


Figure 11 Time series plot of the project manager's forecasted production along the project execution

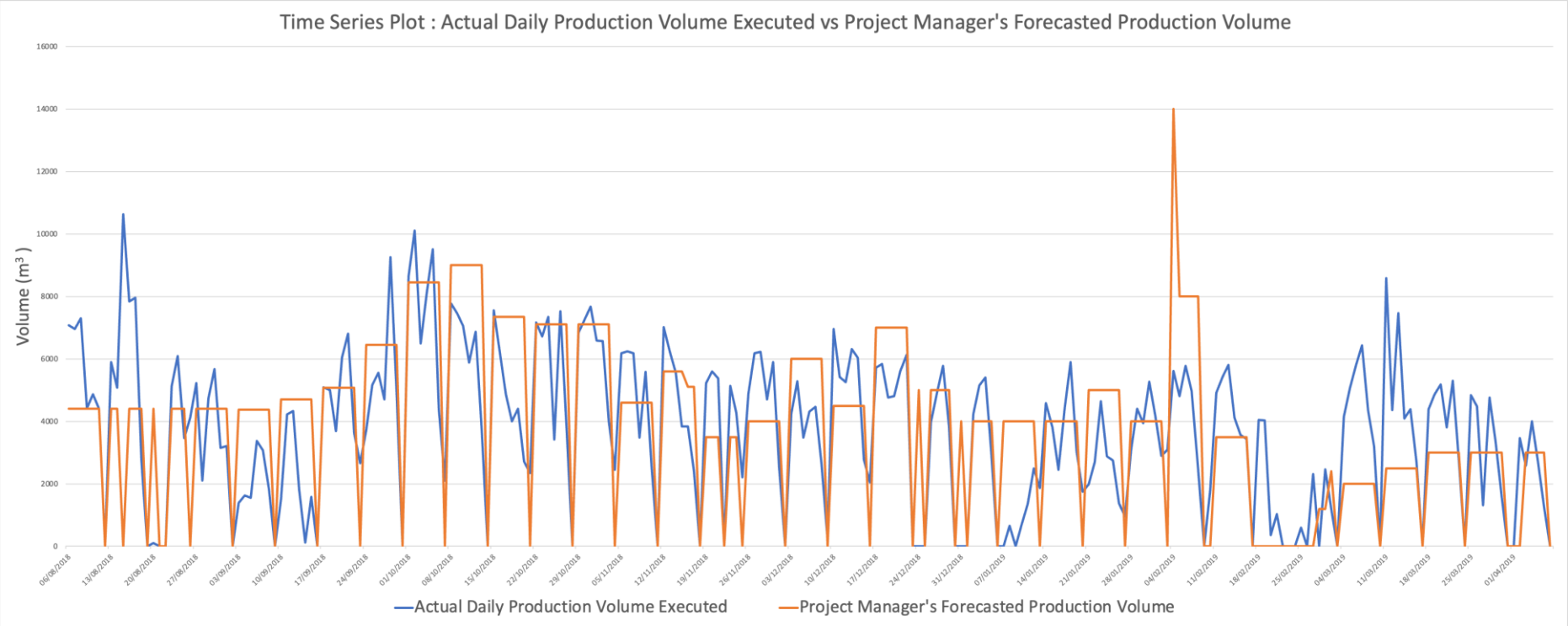


Figure 12 Time series plot of the actual daily volume executed and the project manager's forecasted production volume

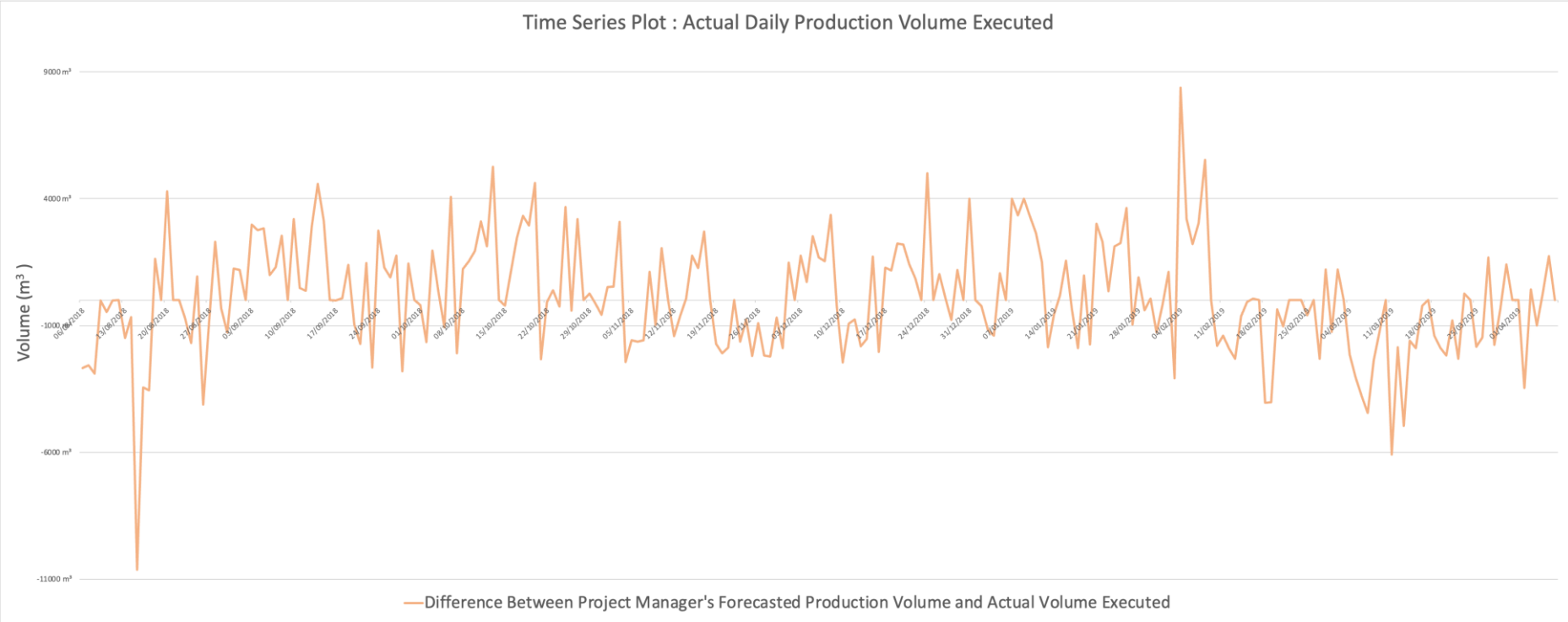


Figure 13 Time series plot of the difference between the project manager's forecasted production volume and actual volume executed

Table 7 Summary Statistics of Production Data

	Total Volume Executed (m3)
count	245
mean	3760.83
std	2445.76
min	0
25%	1860
50%	4035
75%	5424
max	10625

Table 7 presents the summary statistics of the production data. The output variable ranges from 0 m^3 to 10,625 m^3 with a mean of 3760 m^3 and a standard deviation of 2445 m^3. It is evident that the data extends along a wide range of values. Figure 14 shows the histogram of the actual production volume; the data is not normally distributed and seems a bit skewed.

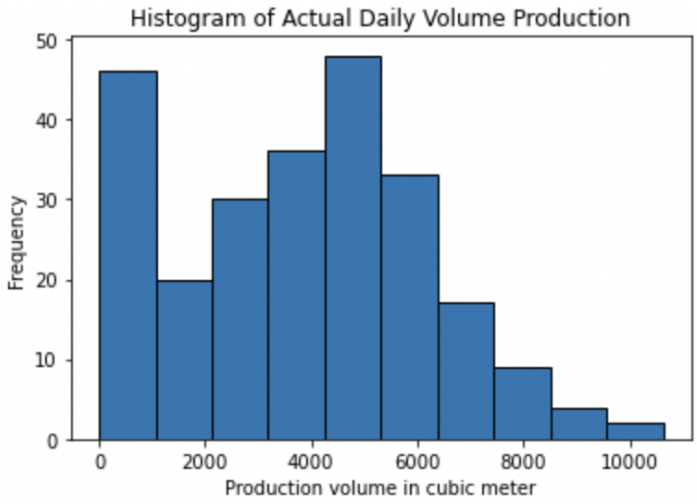


Figure 14 Histogram of excavation volume production output

An ETS (error, trend, seasonal) decomposition was performed in order to visualize and investigate the trend, seasonal, and error patterns that exist within the data.

Figure 15 presents the decomposition of the time series. It is evident that the trend has a changing direction along the execution of the activity. The change in trend can be attributed to changes in project complexity, changes in site conditions, changes in soil conditions, and changes in operational and managerial factors. The use of time series will adapt to the changing behavior in the project over the various phases. The seasonal component is significant to be weekly. The seasonal weekly pattern is related to the weekly routine of the resources where the day of the week may affect the production performance. Every time series plot contains residual and error values that are not considered in the trend and seasonality components. The ANN time series will account for the complex changing behavior and the nonlinearity of the data.

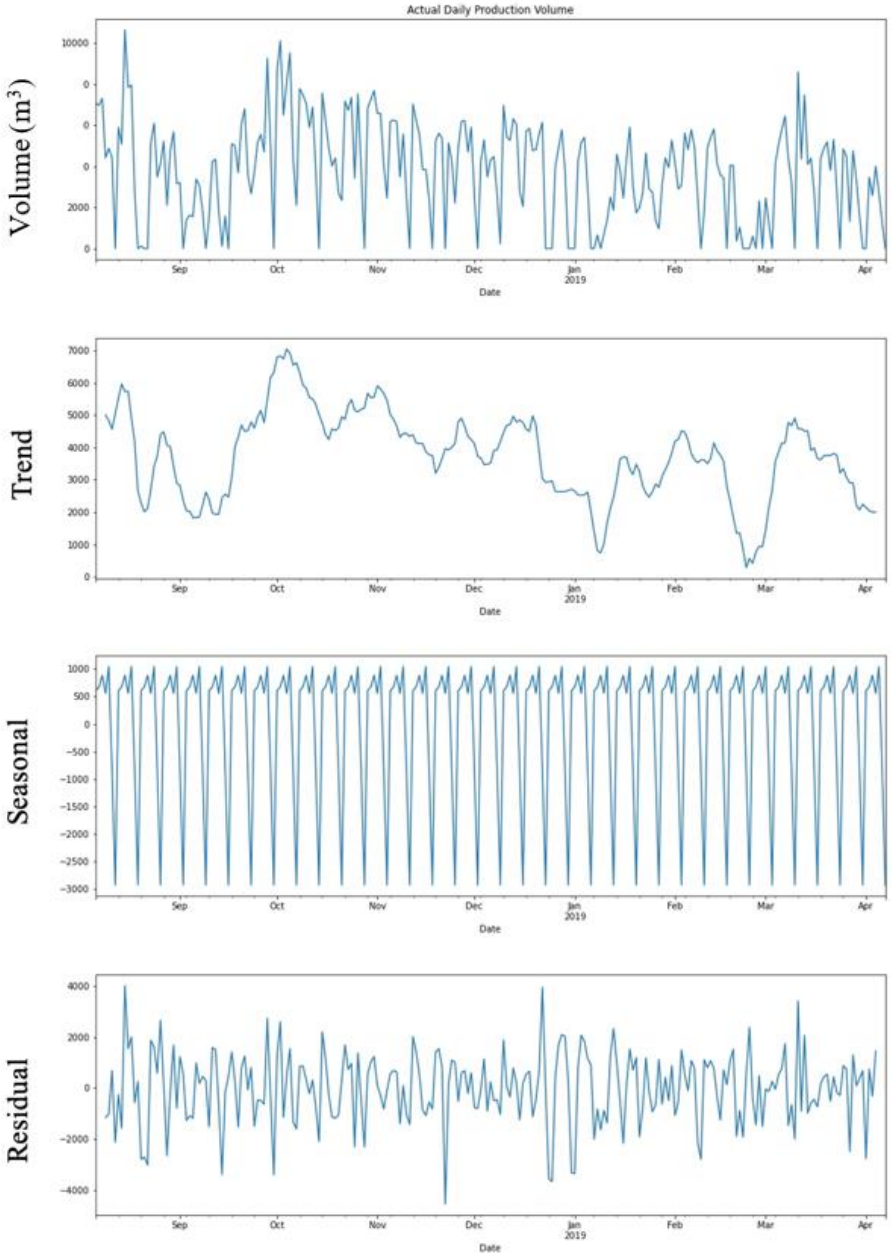


Figure 15 Time Series Plot with decomposition of trend, seasonality, and residual (ETS)

Excavation work is mainly an equipment-intensive activity. Therefore, it is essential to investigate the various equipment being used and select the most relevant ones. The available 23 equipment resources used in this activity are shown in Figure 16. The hours of each equipment were summed for each day. Intuitively, the main resources that drive this activity are the excavators and haulers. Although other several equipment

are used to perform this activity, they are considered to be the support fleet for the main equipment. Consequently, the excavators and haulers will be the only chosen resources to act as input features for the model due to their direct impact and main contribution in affecting the production output. The resource hours were restructured as observed in Table 8. Each column corresponds to the hours of the excavators and haulers used. The various excavators in the project were categorized according to their weight since their weight can reflect their capacity and performance. The excavators used are divided into 4 categories based on their weight in tons: 25 tons, 30 tons, 40 tons, 50 tons, 8 tons. The haulers are divided into 2 categories (A35 and A40) based on their volume capacity. The A35 haulers have a capacity of 14 m^3, while the A40 haulers have a capacity of 16 m^3. The equipment were used interchangeably each day.

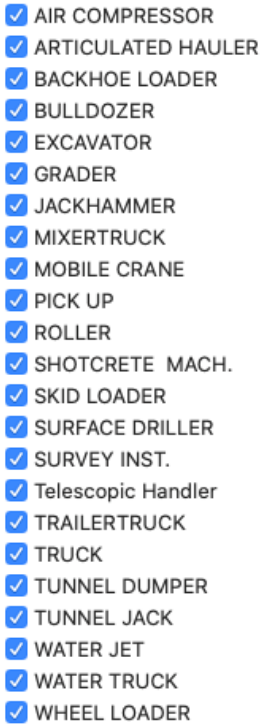


Figure 16 Equipment used on site for the excavation work

Table 8 Structured daily hours of the utilized equipment resources

Date	EXCAVATOR (25 T) Daily Hours	EXCAVATOR (30 T) Daily Hours	EXCAVATOR (40 T) Daily Hours	EXCAVATOR (50 T) Daily Hours	EXCAVATOR (8 T) Daily Hours	Hauler A35 Daily Hours	Hauler A40 Daily Hours
8/6/18	0	0	66	22	0	317	11
8/7/18	0	0	81	23	0	305	11
8/8/18	0	0	68	46	0	374	0
8/9/18	9	0	62	24	0	345	0
8/10/18	8	0	55	44	0	330	0
8/11/18	6	6	30	24	0	142	0
8/12/18	0	0	0	0	0	0	0
8/13/18	0	0	59	42	0	350	0
8/14/18	1	29	57	58	0	420	11
8/15/18	0	22	80	46	0	384	0
8/16/18	0	12	68	47	0	359	11
8/17/18	10	18	55	41	0	290	44
8/18/18	2	6	33	24	0	174	0
8/19/18	0	0	0	0	0	0	0
8/20/18	0	0	12	12	0	42	18

For the weather data, the available weather data contain several parameters that are not relevant to the excavation works. The main weather parameters that can affect the corresponding activity are narrowed down to temperature, humidity, wind speed, and precipitation. These weather factors can directly impact construction performance. Consequently, these selected weather factors are later on tested in the feature selection step to further remove the ones that are not relevant. The remaining weather parameters (dew point, solar radiation, VPD pressure deficit, wind direction, air pressure, solar panel, and battery) were disregarded and dropped since they are not relevant.

Because the available weather data is reported as hourly, the weather data was converted into daily data.

It is important to have reflective weather data of the corresponding day. As a result, the maximum, the average, and the minimum temperature were obtained for each day. The average temperature was obtained by averaging the hourly average temperature from the raw data. The maximum temperature was obtained by getting the maximum hourly temperature along the day, and the minimum temperature was also obtained by getting the minimum hourly temperature along the day. Similarly, the average, maximum, and minimum humidity values were obtained through the same procedure. Also, the

average and maximum wind speed were obtained through the same procedure. The precipitation was obtained by summing the values across every day. Table 9 displays the structured weather parameters.

Table 9 Structured daily weather parameters

Date	Average Temperature (°C)	Max Temperature (°C)	Min Temperature (°C)	Average Humidity (%)	Max Humidity (%)	Min Humidity (%)	Average Wind speed (km/h)	Max Wind speed (km/h)	Total Precipitation (mm)
8/6/18	23.9	27.7	19.3	77.2	94.8	56.1	5.26	10.8	0.6
8/7/18	22.1	27.5	16.6	72.6	95.8	44.0	4.75	10.8	0.0
8/8/18	22.8	28.7	17.3	71.1	95.8	40.0	4.68	10.4	0.0
8/9/18	23.8	29.8	19.1	67.3	93.6	49.4	5.42	10.4	0.0
8/10/18	24.1	30.2	18.6	58.0	89.8	33.6	5.41	10.8	0.0
8/11/18	24.0	31.4	18.2	58.2	91.6	29.4	5.13	10.1	0.0
8/12/18	0.0	0.0	0.0	0.0	0.0	0.0	5.08	9.4	0.0
8/13/18	22.8	29.8	18.0	58.3	90.5	25.2	5.22	11.2	0.0
8/14/18	22.3	28.2	16.7	61.7	92.3	37.0	5.23	10.1	0.0
8/15/18	22.9	29.4	17.4	59.5	92.8	34.0	5.55	10.8	0.0
8/16/18	22.9	29.4	17.6	61.4	96.2	33.1	5.66	10.8	0.0
8/17/18	21.6	27.7	17.1	71.0	94.3	42.8	4.97	11.2	0.0

- Feature Selection

After structuring and selecting the necessary resources and weather parameters, a correlation- based feature selection was performed. The Spearman correlation of the input variables (resource hours and weather parameters) with the output variable (production) was established as shown in the heatmap Spearman correlation matrix in Figure 17. The first column in Figure 17 shows the Spearman correlation between the relevant factors considered with the production output of total volume executed; these values are displayed in Table 10.

Table 10 Spearman correlation of the considered factors with the production output

Factor	Spearman Correlation with Total Volume Executed (m3)
Total Volume Executed (m3)	1.00
Average Temperature (°C)	0.55
Max Temperature (°C)	0.53
Min Temperature (°C)	0.54
Average Humidity (%)	0.33
Max Humidity (%)	0.35
Min Humidity (%)	0.36
Average Wind speed (km/h)	0.11
Max Wind speed (km/h)	0.05
Total Precipitation (mm)	0.11
EXCAVATOR (25 T) Daily Hours	0.10
EXCAVATOR (30 T) Daily Hours	0.34
EXCAVATOR (40 T) Daily Hours	0.71
EXCAVATOR (50 T) Daily Hours	0.70
EXCAVATOR (8 T) Daily Hours	0
Hauler A35 Daily Hours	0.75
Hauler A40 Daily Hours	0.27

For the weather factors, the temperature and humidity parameters show a significant correlation with the total production volume since they range from 0.33-0.55. As a result, these features were kept in the model. It is evident that the average wind speed and max wind speed can be disregarded due to their low correlation that is close to 0. The Spearman correlation of the precipitation with the total production volume is also close to zero; however, this may be due to the fact that the data set contains few data points (85 days out of 245 days) that experienced precipitation instances. The 85 days that experienced precipitation were selected, and their correlation with the total production

volume was obtained. A Spearman correlation of -0.11 is obtained. Precipitation is a major factor affecting production output in excavation activities since it might disrupt the workflow if present. Therefore, the precipitation value was kept in the model although the precipitation has a low correlation with the total production volume.

It is evident that the average, maximum, and minimum temperature features are highly correlated with each other having a correlation close to 1. Consequently, the maximum and minimum temperature features can be dropped. Similarly, the humidity features have the same case, so the average humidity will only be adopted.

As the resource hours increase, the production volume should increase since more hours are being invested in the production process. However, since productivity fluctuates over time, this concept may not apply at all instances. The Pearson correlation of the excavator hours with the total production volume is significant. For instance, the Spearman correlation of the 40 tons excavator with the total production volume is 0.71. However, the 8 tons excavator scored a low correlation of zero with the total production volume because it was rarely used on site, approximately for only 2 weeks in one year. Although the 8 tons has a low correlation with the total production volume, it was kept in the model due to its presence in the excavator groups which is a major resource on site. Similarly, the haulers show a high correlation with the total production volume of 0.75 for the A35 haulers.

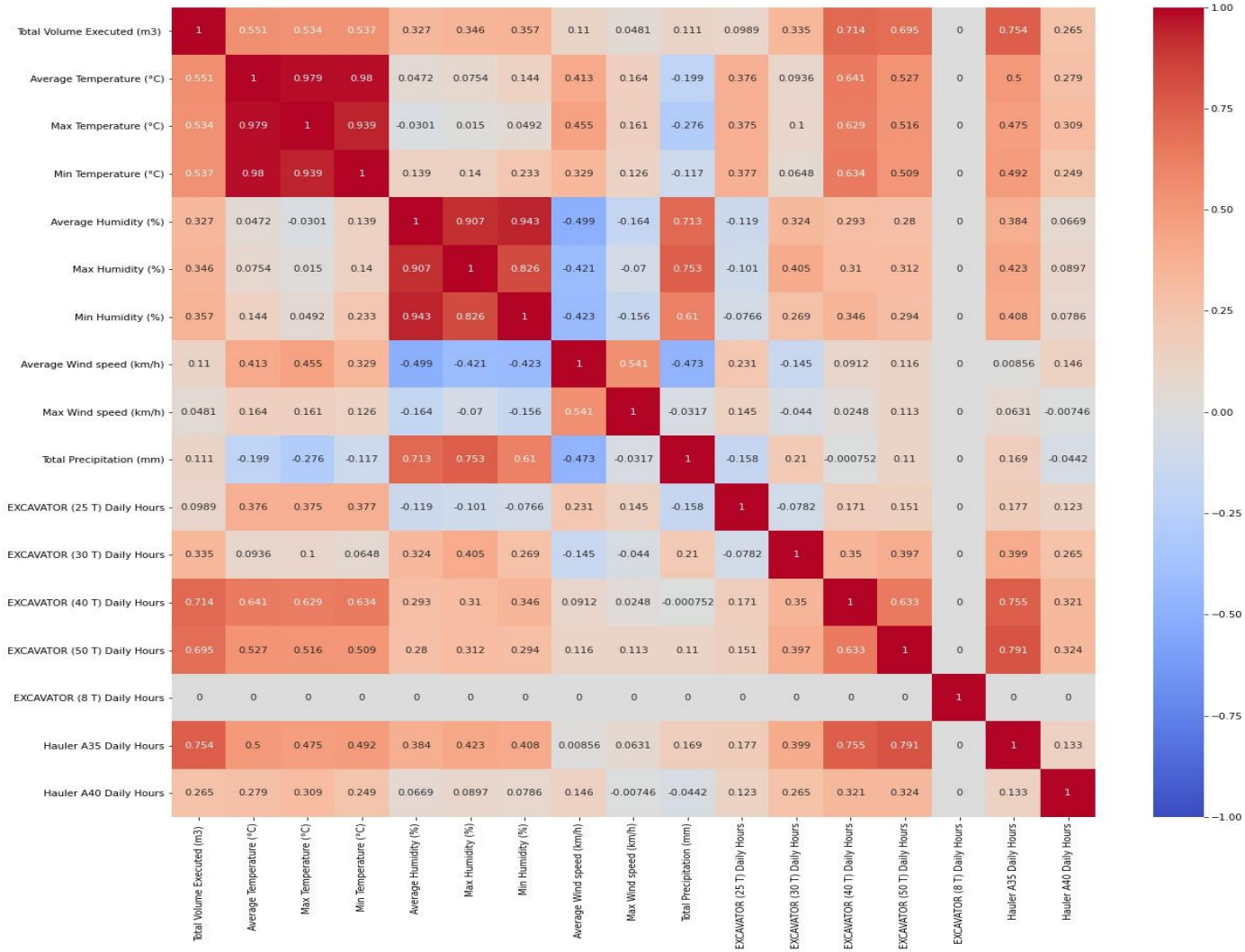


Figure 17 Heatmap of Spearman correlation coefficient matri

3. ANN Data Preparation

a. ANN input identification

At this stage, the following 10 input features are selected: average temperature, average humidity, precipitation, excavator hours (25T, 30T, 40T, 50T, 8T), hauler hours (A35, A40).

b. Feature Scaling

Following that, the min-max feature scaling was performed to transform the data into the range 0 to 1. This is because, ML algorithms, in general, do not perform well when the input numerical features have very different scales (Géron, 2019).

c. Data Split

The data, which consists of 245 data points, was split sequentially into three sets: training, validation, and test set. The training set includes the first 203 data points to fit the model. To evaluate properly the model, the data was split into validation and test set, each having three weeks of data or 21 data points. For this case study, the project manager is interested to forecast the production on a weekly basis, or for the next 7 days, to properly manage the on-site resource. Thus, a multistep forecast of one week, or 7 days, is satisfactory for proper resource planning and production estimation of excavation activities, so the required forecast horizon is 7 days. As previously mentioned, in time series forecasting models, the validation and test sets can have the size of the maximum required forecast horizon; hence, the validation and test set size can be 7 days each. However, for a more conservative approach, 3 sets of 7 days or 3 weeks were selected for each of the validation and test set, producing a total of 21 datapoints for each set. In the

upcoming steps, the walk-forward approach is adopted on the validation and test set where the multi-step weekly forecast is performed on each set. The multistep forecast for validation and testing is established in a recursive manner where the predictions of the first step is used as input for the consecutive forecast. Figure 18 displays the data split of the training, validation, and test set. The other corresponding variables were also split the same approach.

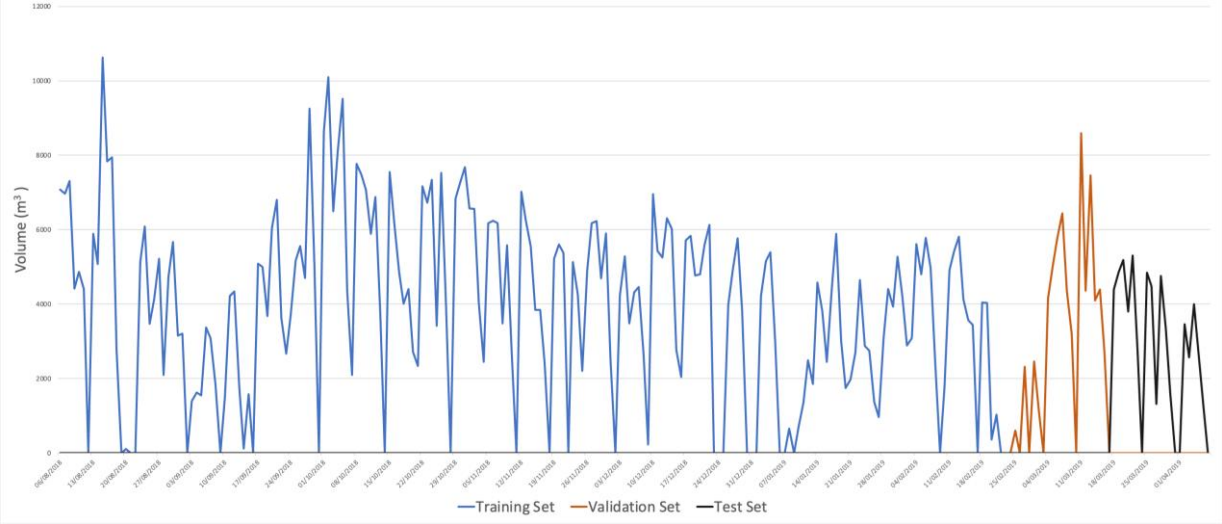


Figure 18 Data split for time series model

4. ANN Model Development

a. MLP time series model

The MLP ANN was developed in Python using the Keras library. When developing the model, various potential models were tested and evaluated, but several ones failed to meet the required evaluation conditions of training, validation, stability, and testing requirements. Consequently, several other models were tested by modifying

the various hyperparameters in a systematic way. A suggested model is presented in the following sections.

A lag of 7 days was selected for this model. The spearman correlation test shows a significant correlation of 0.6 between the 7-day lag of production output. Also, a 7 days lag interval is reasonable due to the presence of weekly patterns in the data as previously discussed in the statistical model.

In other words, the model uses the previous production output of the 7 days as input to perform a prediction. Also, it uses as input the exogenous factors that are the weather features (3) and resource hours (7) of the corresponding day. The total number of input features becomes 17.

The network architecture of the MLP model is comprised of one input layer, one hidden layer, and one output layer. In the input layer, the number of neurons is equivalent to the number of input features (17). In the hidden layer, 80 neurons were used. For the output layer, only one neuron is designated since it is a regressions problem where the production value is being forecasted.

The RELU activation function was used in the hidden layer due to its superior performance. Also, the Adam optimizer was used for the training algorithm since it has a superior performance in the literature and in the model performance. The default values were used for the parameters $\beta 1$ (0.9) and $\beta 2$ (0.999). A learning rate η of 0.001 was selected. 120 number of training epochs was selected for building the model. The MSE and RMSE function was adopted to evaluate the performance of the model. To ensure that the model is not prone to overfitting problems, the Ridge ($\ell 2$) and Lasso ($\ell 1$) regularization methods were adopted. The selected Ridge and Lasso hyperparameters were respectively 0.000001 and 0.000001 for the MLP

time lag model. The following hyperparameters were being updated between every model iteration: the number of lags with the corresponding number of input features, number of neurons in the input layer and each hidden layer, number of hidden layers, regularization method, number of training epoch, and the learning rate.

Table 11 below summarizes the selected hyperparameters for the MLP time series model. The selected hyperparameters were used to fit and evaluate the model. Table 12 summarizes the results of the MLP time series model.

Table 11 MLP time series model hyperparameters and characteristics

	MLP time series model
Number of lags	7 days
Network architecture:	
Number of hidden layers	1
Number of neurons in input layer	17
Number of neurons in each hidden layer	80
Number of neurons in output layer	1
Network activation function:	
Activation function in hidden layer (s)	ReLu
Activation function in output layer	Linear
Optimization algorithm:	Adam
Momentum decay hyperparameter: $\beta 1$	0.9
Scaling decay hyperparameter: $\beta 2$	0.999
Learning rate: η	0.001
Training batch size	
Number of training epoch	120
Performance function	MSE
Regularization:	
Ridge regularization $\ell 1$	0.000001
Lasso regularization $\ell 2$	0.000001

Table 12 MLP time series model error metrics summary results

	MSE	RMSE	R^2
Training Set	662596	814	0.89
Validation Set	1052676	1026	0.85
Test Set	687536	829	0.81

The model has a RMSE of 814 m^3 on the training set and an R^2 of 0.89. Given the context of the problem and the magnitude of the production value ranging from 0 to 10,000 m^3, the error seems to be minimal and acceptable. Also, the model is fitting well the data points since it can explain 89% of the training dataset. Figure 19 shows the predicted and actual values of the production values for the training set using the MLP time series model.

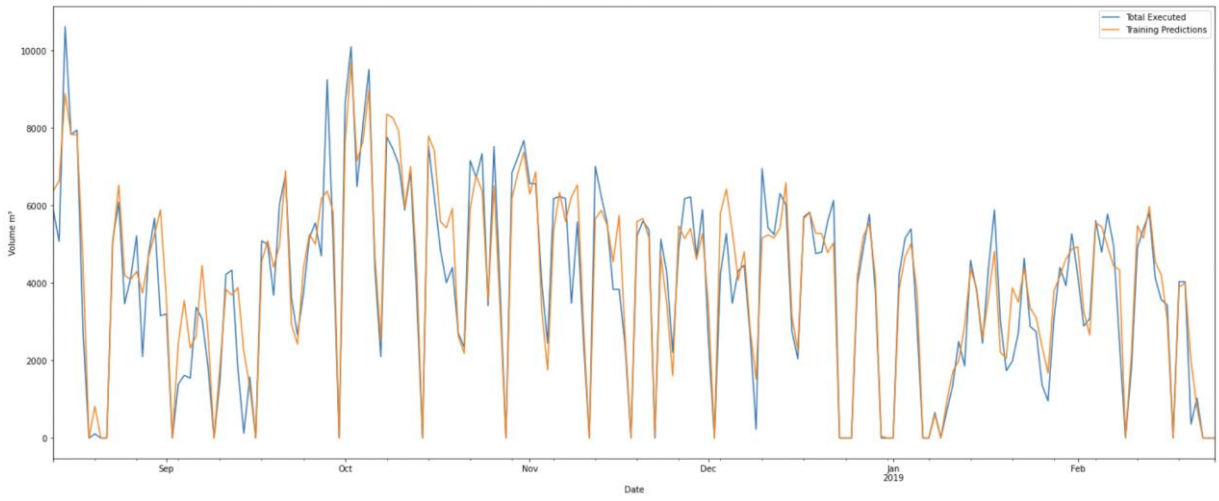


Figure 19 MLP time series model actual and predicted values in the train set

Using the walk-forward validation, the model was evaluated on the validation set. A RMSE of 1,026 m^3 was obtained along with a R^2 of 0.85. The validation RMSE is approximately close to the RMSE of the training set; thus, the model is not overfitting. Also, the fact that the validation error is a bit higher than the training error signifies that

the model is valid because the model is anticipated to underperform on the data which was not used for fitting. In addition, the model is fitting well the validation data points by explaining 85% of the data. Figure 20 shows the predicted and the actual values of the production for the validation set using the MLP time series model.

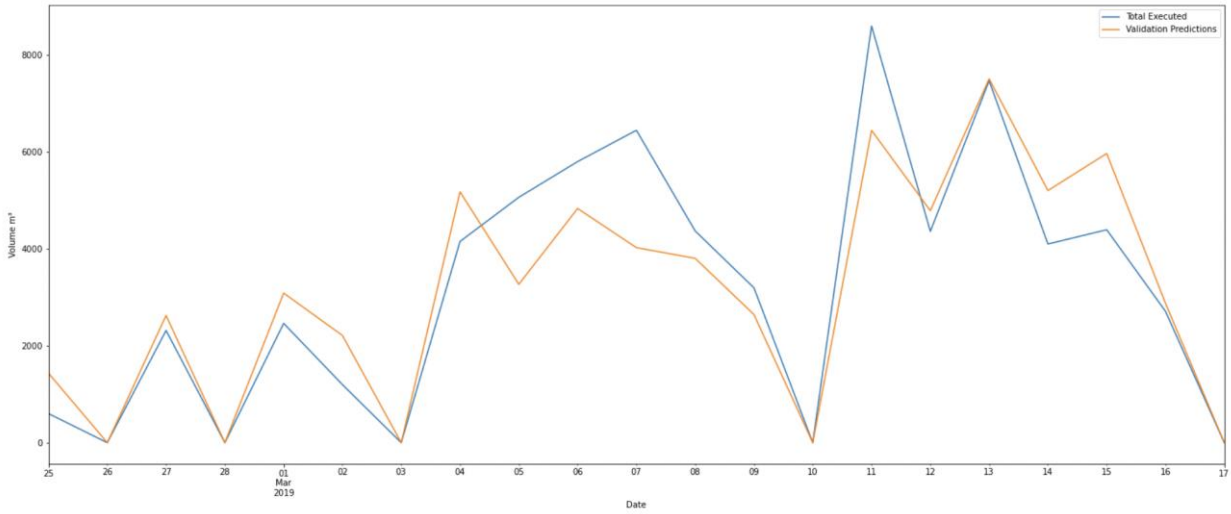


Figure 20 MLP time series model actual and predicted values in the validation set

Due to the stochastic nature of the MLP models, the model was fit 30 times using the same hyperparameters to ensure its stability. Through every iteration, the initial weights is the only parameter that is changing because of the stochastic nature of the neural network optimization algorithm. The mean average for the train RMSE is 840 m^3 with a standard deviation of 40. Also, the mean average of the validation RMSE was 1100 m^3 and a standard deviation of 60. The model is considered stable since the average train and validation errors are close to the selected model's corresponding train and validation errors. Also, both have a low standard deviation. Thereby, the model's hyperparameters show adequate results.

The model was further evaluated on the test set using the walk-forward validation. A test set RMSE of 829 m^3 was obtained along with a R^2 of 0.81. The error is considered

to be low, and it is slightly higher than the train RMSE; hence, the model is not overfitting. The results show an adequate fit since the model is explaining 81% of the data. Figure 21 shows the predicted and the actual values of the production for the test set using the MLP time series model.

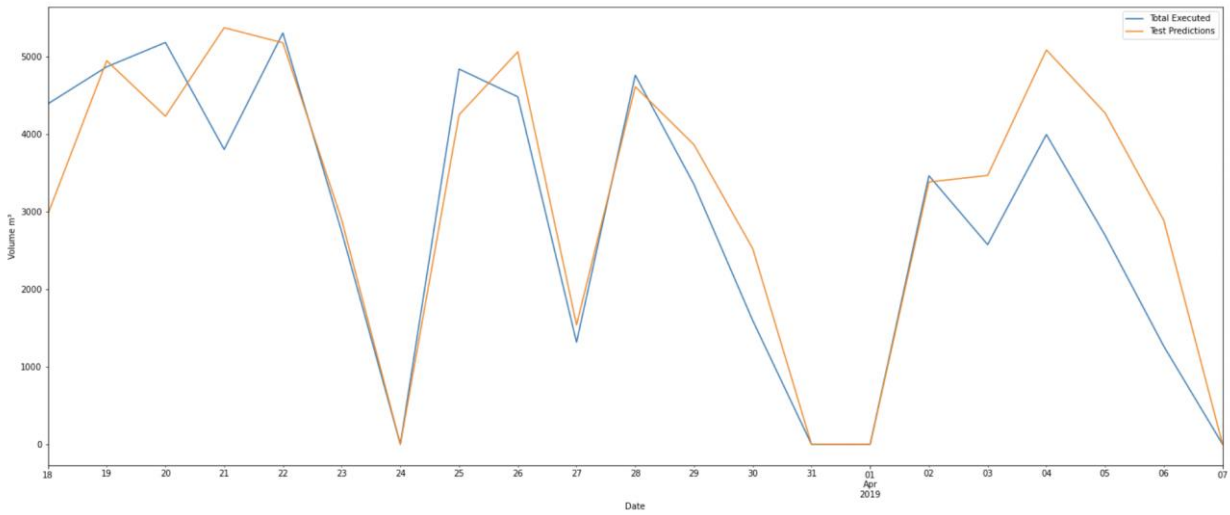


Figure 21 MLP time series model actual and predicted values in the test set

b. LSTM

The LSTM was developed in Python using the Keras library. When developing the model, various potential models were tested and evaluated, but several ones failed to meet the required evaluation conditions of training, validation, stability, and testing requirements. Consequently, several other models were tested by modifying the various hyperparameters in a systematic way. A suggested model is presented in the following sections.

Although the LSTM model uses the sequence of the data and captures the temporal dependencies present within the data, its performance can be improved by choosing a time lag as a predictor (Bouktif et al., 2019). After various iterations, a lag of

14 days was selected since it shows a superior performance compared with other time lags.

In other words, the model uses the previous production output of the 14 days as input to perform a prediction. Also, it uses as input the exogenous factors that are the weather features (3) and resource hours (7) of the corresponding day. The total number of input features becomes 24.

The network architecture of the LSTM model is comprised of one input layer, one hidden layer, and one output layer. In the input layer, the number of neurons is equivalent to the number of input features (24). In the hidden layer, 50 neurons were used. For the output layer, only one neuron is designated since it is a regressions problem where the production value is being forecasted.

The RELU activation function was used in the hidden layer due to its superior performance. Also, the Adam optimizer was used for the training algorithm since it proves a superior performance in the literature and in the model performance. The default values were used for the parameters $\beta 1$ (0.9) and $\beta 2$ (0.999). A learning rate η of 0.001 was selected. 110 number of training epochs was selected for building the model. The MSE and RMSE function was adopted to evaluate the performance of the model. To ensure that the model is not prone to overfitting problems, the Ridge ($\ell 2$) and Lasso ($\ell 1$) regularization methods were adopted. The selected Ridge and Lasso hyperparameters were respectively 0.000001 and 0.000001 for the LSTM model. The following hyperparameters were being updated between every model iteration: the number of lags with the corresponding number of input features, number of neurons in the input layer and each hidden layer, number of hidden layers, regularization method, number of training epoch, and the learning

rate.

Table 13 below summarizes the selected hyperparameters for the LSTM model.

Table 13 LSTM model hyperparameters and characteristics

	LSTM model
Number of lags	14 days
Network architecture:	
Number of hidden layers	1
Number of neurons in input layer	24
Number of neurons in each hidden layer	50
Number of neurons in output layer	1
Network activation function:	
Activation function in hidden layer (s)	ReLu
Activation function in output layer	Linear
Optimization algorithm:	Adam
Momentum decay hyperparameter: $\beta 1$	0.9
Scaling decay hyperparameter: $\beta 2$	0.999
Learning rate: η	0.001
Training batch size	
Number of training epoch	110
Performance function	MSE
Regularization:	
Ridge regularization $\ell 1$	0.000001
Lasso regularization $\ell 2$	0.000001

The selected hyperparameters were used to fit and evaluate the model. Table 14 summarizes the results of the LSTM model.

Table 14 LSTM model error metrics summary results

	MSE	**RMSE**	**R^2**
Training Set	723707	851	0.87
Validation Set	1463751	1210	0.78
Test Set	756878	870	0.77

The model has a RMSE of 851 m^3 on the training set and an R^2 of 0.87. Given the context of the problem and the magnitude of the production value ranging from 0 to 10,000 m^3, the error seems to be minimal and acceptable. Also, the model is fitting well the data points since it can explain 87% of the training set. Figure 22 shows the predicted and actual values of the production values for the training set using the LSTM model.

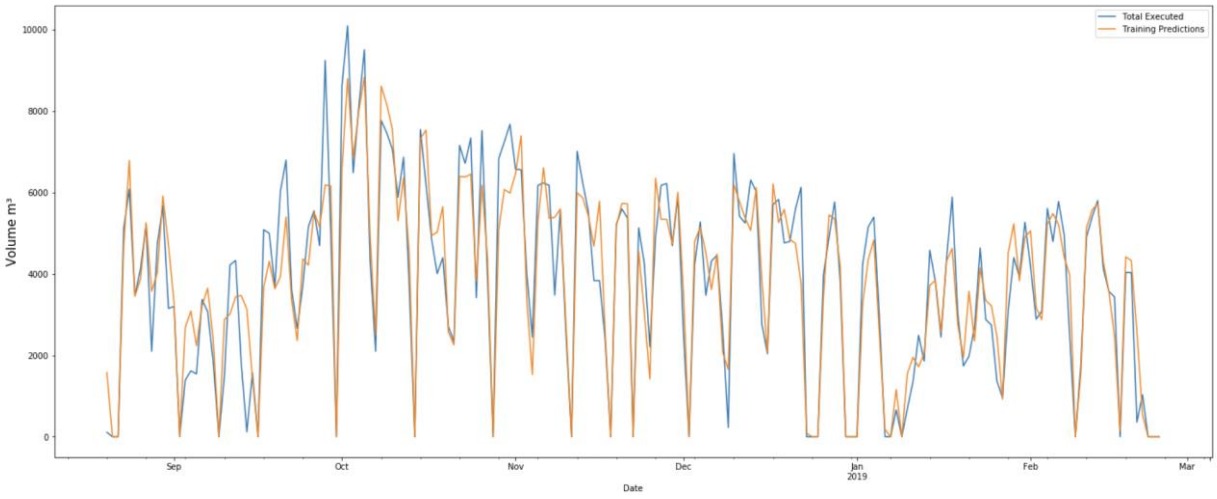


Figure 22 LSTM model actual and predicted values in the train set

Using the walk-forward validation, the model was evaluated on the validation set. A RMSE of 1,210 m^3 was obtained along with a R^2 of 0.78. The validation RMSE is approximately close to the RMSE of the training set; thus, the model is not overfitting. Also, the fact that the validation error is a bit higher than the training error signifies that the model is valid because the model is anticipated to underperform on the data which was not used for fitting. In addition, the model is fitting well the validation data points by explaining 78% of the data. Figure 23 shows the predicted and the actual values of the production for the validation set using the LSTM model.

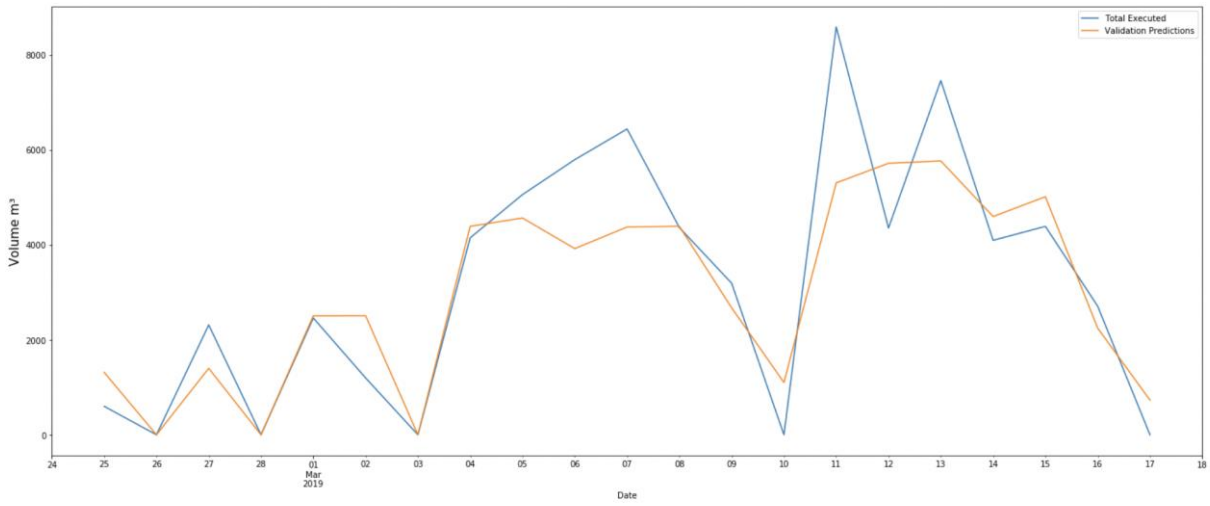


Figure 23 LSTM model actual and predicted values in the validation set

Due to the stochastic nature of the MLP models, the model was fit 30 times using the same hyperparameters to ensure its stability. Through every iteration, the initial weights is the only parameter that is changing because of the stochastic nature of the neural network optimization algorithm. The mean average for the train RMSE is 873 m^3 with a standard deviation of 45. Also, the mean average of the validation RMSE was 1,290 m^3 and a standard deviation of 73. The model is considered to be stable since the average train and validation errors are close to the selected model's corresponding train and validation errors. Also, both have a low standard deviation. Thereby, the model's hyperparameters show adequate results.

The model was further evaluated on the test set using the walk-forward validation. A test set RMSE of 870 m^3 was obtained along with a R^2 of 0.77. The error is considered to be low, and it is slightly higher than the train RMSE; hence, the model is not overfitting. The results show an adequate fit since the model is explaining 77% of the data. Figure 24 shows the predicted and the actual values of the production for the test set using the LSTM model.

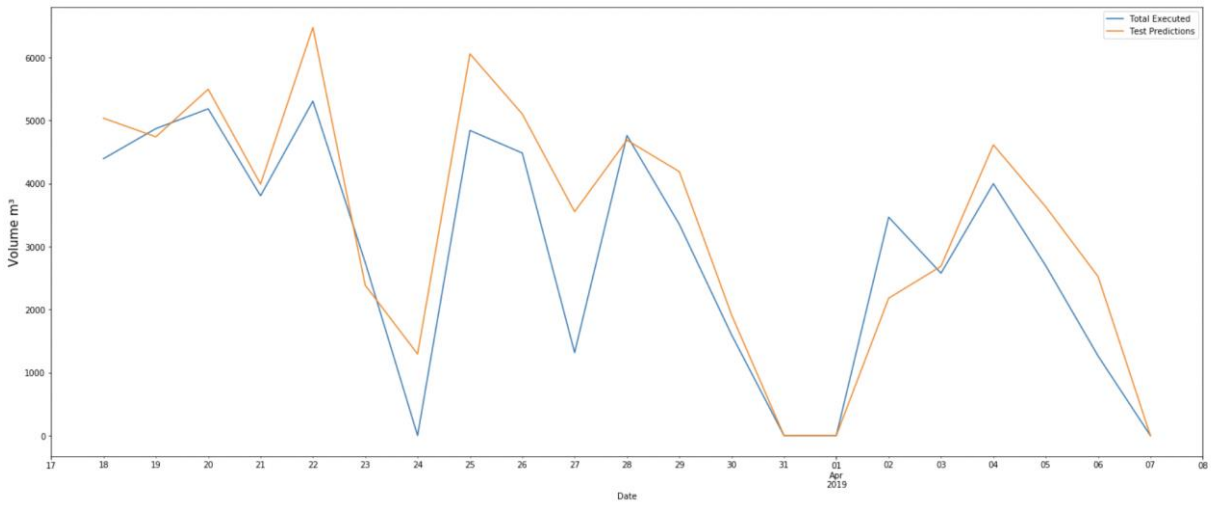


Figure 24 LSTM model actual and predicted values in the test set

c. GRU

The GRU was developed in Python using the Keras library. When developing the model, various potential models were tested and evaluated, but several ones failed to meet the required evaluation conditions of training, validation, stability, and testing requirements. Consequently, several other models were tested by modifying the various hyperparameters in a systematic way. A suggested model is presented in the following sections.

Similar to the LSTM model, choosing a time lag as a predictor improves the prediction performance of the model (Bouktif et al., 2019). A lag of 7 days was selected. The spearman correlation test shows a significant correlation of 0.6 between the 7 days lag of production output. Also, a 7-day lag interval is reasonable due to the presence of weekly patterns in the data as previously discussed in the statistical model.

In other words, the model uses the previous production output of the 7 days as input to perform a prediction. Also, it uses as input the exogenous factors that are the weather

features (3) and resource hours (7) of the corresponding day. The total number of input features becomes 17.

The network architecture of the GRU model is comprised of one input layer, one hidden layer, and one output layer. In the input layer, the number of neurons is equivalent to the number of input features (17). In the hidden layer, 80 neurons were used. For the output layer, only one neuron is designated since it is a regression problem where the production value is being forecasted.

The RELU activation function was used in the hidden layer due to its superior performance. Also, the Adam optimizer was used for the training algorithm since it proves a superior performance in the literature and in the model performance. The default values were used for the parameters $\beta 1$ (0.9) and $\beta 2$ (0.999). A learning rate η of 0.001 was selected. 120 number of training epochs was selected for building the model.

The MSE and RMSE function was adopted to evaluate the performance of the model. To ensure that the model is not prone to overfitting problems, the Ridge ($\ell 2$) and Lasso ($\ell 1$) regularization methods were adopted. The selected Ridge and Lasso hyperparameters were respectively 0.000001 and 0.000001 for the GRU model.

The following hyperparameters were being updated between every model iteration: the number of lags with the corresponding number of input features, number of neurons in the input layer and each hidden layer, number of hidden layers, regularization method, number of training epoch, and the learning rate.

Table 15 below summarizes the selected hyperparameters for the GRU model.

Table 15 GRU model hyperparameters and characteristics

	GRU model
Number of lags	7 days
Network architecture:	
Number of hidden layers	1
Number of neurons in input layer	17
Number of neurons in each hidden layer	80
Number of neurons in output layer	1
Network activation function:	
Activation function in hidden layer (s)	ReLu
Activation function in output layer	Linear
Optimization algorithm:	Adam
Momentum decay hyperparameter: $\beta 1$	0.9
Scaling decay hyperparameter: $\beta 2$	0.999
Learning rate: η	0.001
Training batch size	
Number of training epoch	120
Performance function	MSE
Regularization:	
Ridge regularization $\ell 1$	0.000001
Lasso regularization $\ell 2$	0.000001

The selected hyperparameters were used to fit and evaluate the model. Table 16 summarizes the results of the GRU model.

Table 16 GRU model error metrics summary results

	MSE	**RMSE**	**R^2**
Training Set	791897	890	0.87
Validation Set	1463751	1209	0.78
Test Set	818536	904	0.75

The model has a RMSE of 890 m^3 on the training set and an R^2 of 0.87. Given the context of the problem and the magnitude of the production value ranging from 0 to 10,000 m^3, the error seems to be minimal and acceptable. Also, the model is fitting well the data

points since it can explain 87% of the training set. Figure 25 shows the predicted and actual values of the production values for the training set using the GRU model.

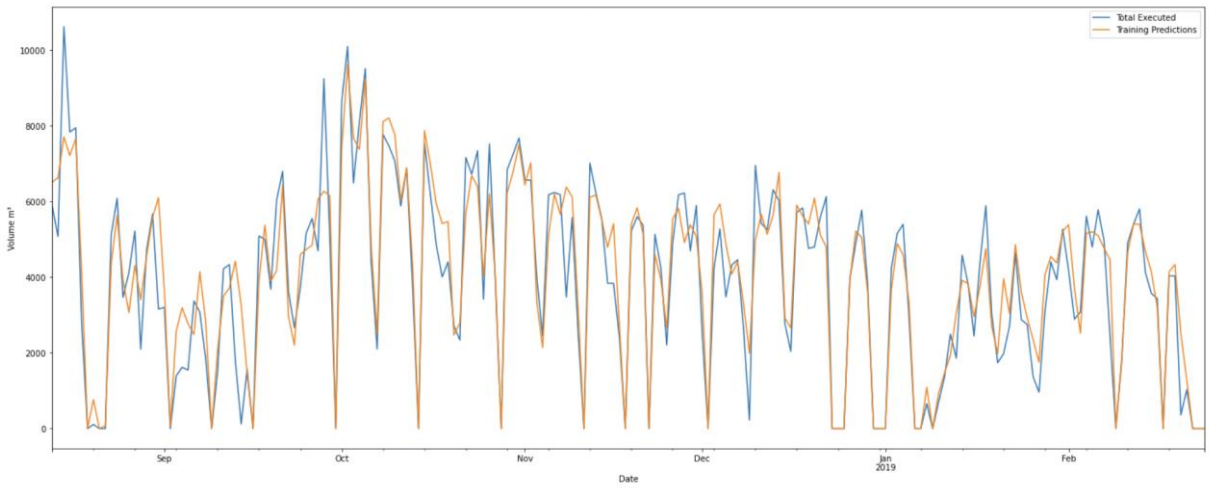


Figure 25 GRU model actual and predicted values in the train set

Using the walk-forward validation, the model was evaluated on the validation set. A RMSE of 1,209 m^3 was obtained along with a R^2 of 0.78. The validation RMSE is approximately close to the RMSE of the training set; thus, the model is not overfitting. Also, the fact that the validation error is a bit higher than the training error signifies that the model is valid because the model is anticipated to underperform on the data which was not used for fitting. In addition, the model is fitting well the validation data points by explaining 78% of the data. Figure 26 shows the predicted and the actual values of the production for the validation set using the GRU model.

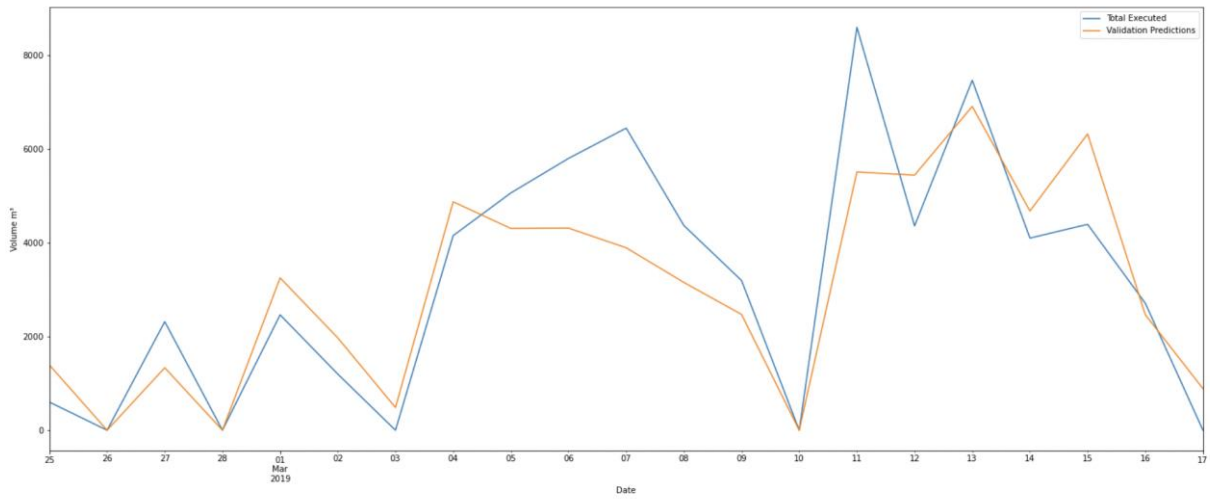


Figure 26 GRU model actual and predicted values in the validation set

Due to the stochastic nature of the MLP models, the model was fit 30 times using the same hyperparameters to ensure its stability. Through every iteration, the initial weights is the only parameter that is changing because of the stochastic nature of the neural network optimization algorithm. The mean average for the train RMSE is 910 m^3 with a standard deviation of 47. Also, the mean average of the validation RMSE was 1270 m^3 and a standard deviation of 82. The model is considered to be stable since the average train and validation errors are close to the selected model's corresponding train and validation errors. Also, both have a low standard deviation. Thereby, the model's hyperparameters show adequate results.

The model was further evaluated on the test set using the walk-forward validation. A test set RMSE of 904 m^3 was obtained along with a R^2 of 0.75. The error is considered to be low, and it is slightly higher than the train RMSE; hence, the model is not overfitting. The results show an adequate fit since the model is explaining 75% of the data. Figure 27 shows the predicted and the actual values of the production for the test set using the GRU model.

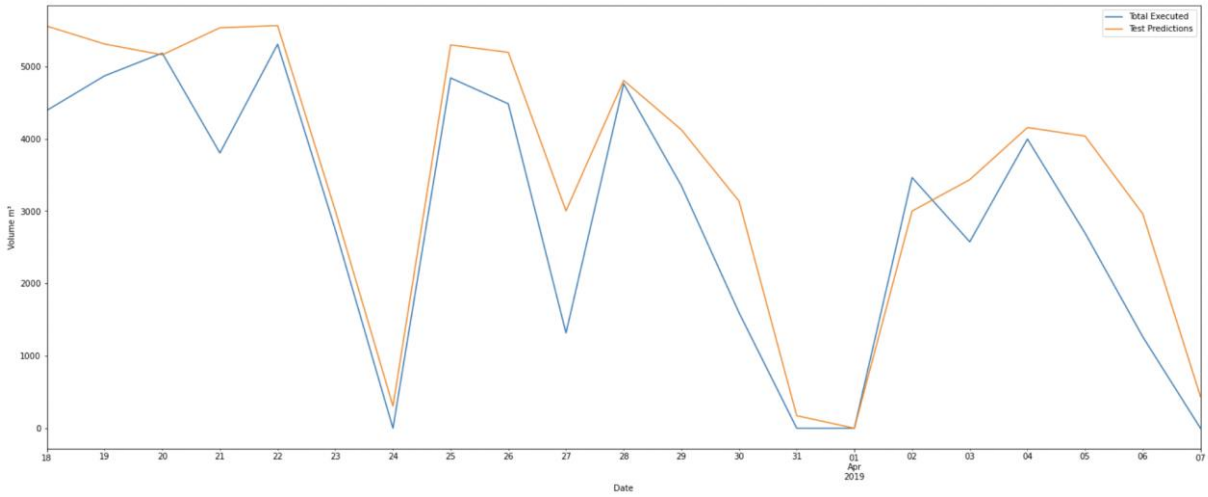


Figure 27 GRU model actual and predicted values in the test set

C. Benchmark Model Development

1. Statistical Time Series

a. Data Collection and Preparation

The same steps and results apply as the ones found in the previous section Data Collection of the generic framework model. For the data preparation and preprocessing, The same steps and results apply as the ones found in the previous section Data Preprocessing of the generic framework model.

b. Stationary Time Series

The first step in fitting an ARIMA model is to determine the differencing order needed to make the series stationary. As previously mentioned, the series must be stationary in the ARIMA model. To test for the stationarity of the series, the augmented dicky fuller (ADF) test was performed on the selected 203 training data sample. The series was determined to be non-stationary since the p-value was 0.35 which is higher than the required 1% significance level to reject the hypothesis.

```
Augmented Dickey-Fuller Test:
ADF test statistic       -1.845884
p-value                   0.357931
# lags used              14.000000
# observations          195.000000
critical value (1%)      -3.464337
critical value (5%)      -2.876479
critical value (10%)     -2.574733
Weak evidence against the null hypothesis
Fail to reject the null hypothesis
Data has a unit root and is non-stationary
```

Figure 28 ADF test results on the original data series

The stationarity of the series can also be detected by visualizing the ACF plots of the series. Figure 29 shows that the ACF plot gradually decays to zero with positive correlations at a high number of lags. Both the ADF test and the ACF plots indicate that this time series is non-stationary. In addition, seasonality can be detected from the ACF plots. The spikes at every 7-day lag interval in the ACF plots shows that there is weekly seasonality in the series. Given these conditions, the data must be transformed to a stationary series by using the differencing method in order to remove the trend and seasonality. Due to the presence of seasonality spikes at every 7-day lag interval, the seasonal integration order "*D*" should be determined in addition to the non-seasonal integration order "*d*".

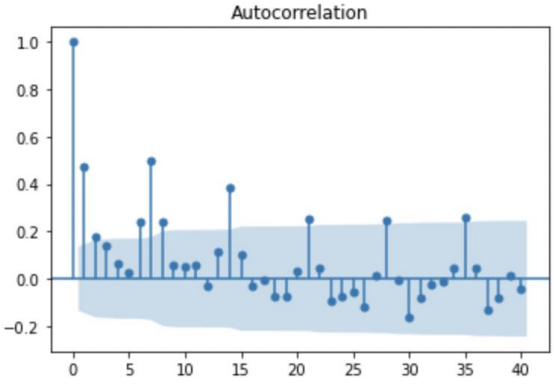


Figure 29 ACF plot of the original data series

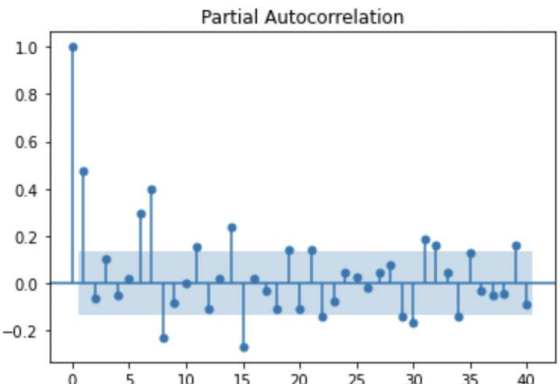


Figure 30 PACF plot of the original data series

Due to the presence of weekly seasonality and the non-stationary behavior of the series, the seasonal differencing order "*D*" was selected to be 1. Consequently, the ADF test results in Figure 31 shows that the p-value is lower than 0.01. So, by taking this seasonal differencing order, the data is transformed into a stationary series. Also, the ACF plot decays directly to zero, thereby indicating a more stationary series. To prevent over differencing, taking only a seasonal differencing order is adequate for this data to make it stationary.

```
Augmented Dickey-Fuller Test:
ADF test statistic     -1.508877e+01
p-value                 8.251084e-28
# lags used             1.500000e+01
# observations          1.870000e+02
critical value (1%)    -3.465812e+00
critical value (5%)    -2.877123e+00
critical value (10%)   -2.575077e+00
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
```

Figure 31 ADF test result of the differenced data series

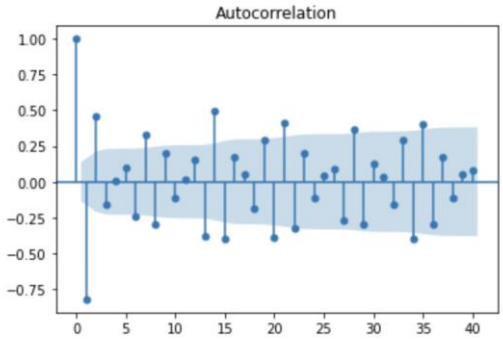


Figure 32 ACF plot of the differenced data series

c. ARIMA (*p*,*d*,*q*) (*P*,*D*,*Q*) *m* development

i. Identification

Given these conditions, the ACF and PACF plots of the transformed series were investigated to identify the initial model parameters. The ACF and PACF plots can give us an initial estimate of the model parameters since they provide signals of the time series.

However, the selection of the model parameters from the ACF and PACF can be misleading. To determine the more suitable model for the given data, it is more advisable to perform various parameter combinations to assess the model's performance (Nau, 2016).

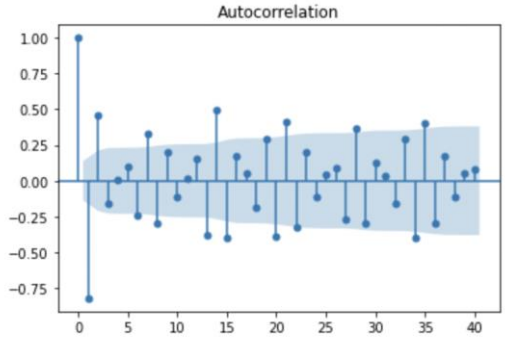


Figure 33 ACF plot of the differenced data series

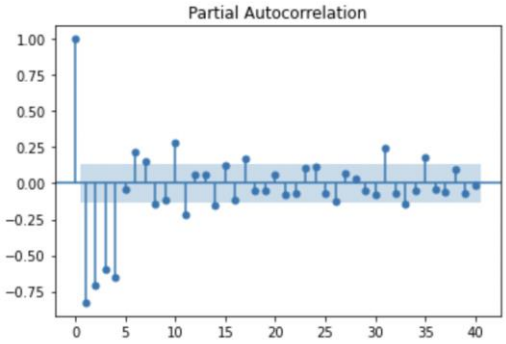


Figure 34 PACF plot of the differenced data series

As an initial model parameter identification, the AR order "*p*", which must be determined from the PACF plots, could range from 0 to 4. In addition, an initial value for the MA order "*q*" parameter could range from 0 to 2 based on the ACF plots. Interpreting theses plots to find the best order can be troublesome. For both orders, a range of 0 to 4 are selected and tested in the upcoming "estimation" step. From the previous step, the integration order "*d*" is selected to be "0", and the seasonal integration order "*D*" is selected to be 1.

SARIMA model parameters also needed to be estimated since the ACF plots signify a weekly seasonality due to the spikes at every 7-day lag interval. The "*m*" value is specified to be 7 in order to consider the weekly seasonality. The seasonality parameters "*P*" and "*Q*" are estimated based on the behavior of the seasonal spikes in the ACF and PACF plots. The "*P*" parameter, that is identified based on the PACF plots, could range from 0 to 4 since significant spikes are present at each 7-day lag interval (i.e.: lags 7, 14, 21, 28, and 35) in the plot. Also, the "*Q*" parameter could range from 0 to 2 based on the ACF plots due to the presence of significant spikes at the 7-day lag interval (i.e.: lags 7 and 14).

As previously mentioned, the identification of the parameters using the ACF and PACF plots can sometimes be tricky and misleading (Nau, 2016); therefore, the performance of various models was assessed. A grid search was performed where several parameter orders were specified to test the various order combinations. The exogenous variables that were previously used in the neural network models were also included in this model, and these include the following 10 features: average temperature, average humidity, precipitation, excavator hours (25T, 30T, 40T, 50T, 8T), hauler hours (A35, A40).

ii. Estimation

Once the model parameters are identified, the estimation step proceeds. The model training performance metric errors (i.e.: MSE, RMSE, R^2) along with the AIC were recorded in every iteration. Table 17 summarizes the results obtained.

Table 17 ARIMA model iterations for the estimation step

Model number	**(*p, d, q*)**	**(*P, D, Q, m*)**	**AIC**	**RMSE train**	**R^2 train**
1	(0, 0, 4)	(2, 1, 4, 7)	2817	1308	0.717
2	(2, 0, 4)	(2, 1, 4, 7)	2820	1341	0.702
3	(1, 0, 4)	(1, 1, 4, 7)	2820	1347	0.699
4	(3, 0, 4)	(2, 1, 4, 7)	2820	1343	0.701
5	(0, 0, 2)	(2, 1, 4, 7)	2870	1346	0.694
6	(1, 0, 2)	(0, 1, 2, 7)	3110	1382	0.683
7	(2, 0, 2)	(0, 1, 2, 7)	3111	1384	0.683
8	(1, 0, 2)	(1, 1, 2, 7)	3112	1370	0.689
9	(2, 0, 2)	(1, 1, 2, 7)	3114	1389	0.681

Good models are obtained by minimizing the AIC and having satisfactory performance metrics. By comparing the AIC of the model and considering the performance metric errors, model 1 was chosen since it represents a better fit with superior performance for the corresponding data. Table 18 shows the coefficients of the variables used in the selected model.

Table 18 ARIMA Model Variable Coefficients

	Factor	Coefficient
ARIMA Orders	MA1	0.3952
	MA2	0.1801
	MA3	0.248
	MA4	0.1556
Seasonal Orders	AR7	-0.96
	AR14	-0.62
	MA7	0.26
	MA14	-0.22
	MA21	-0.66
	MA28	-0.22
Exogenous Factors	Average Temperature (°C)	76.82
	Average Humidity (%)	3.55
	Precipitation (mm)	0.02
	EXCAVATOR (8 T) Daily Hours	-5.749 e-15
	EXCAVATOR (25 T) Daily Hours	-16.88
	EXCAVATOR (30 T) Daily Hours	-2.6
	EXCAVATOR (40 T) Daily Hours	10.2
	EXCAVATOR (50 T) Daily Hours	7.97
	Hauler A35 Daily Hours	8.39
	Hauler A40 Daily Hours	7.63

The model has an RMSE of 1308 m^2 and an R^2 of 0.71. The model was only able to explain 71% of the training data set. Figure 35 shows the predicted and actual values of the production values for the training set using the SARIMAX model.

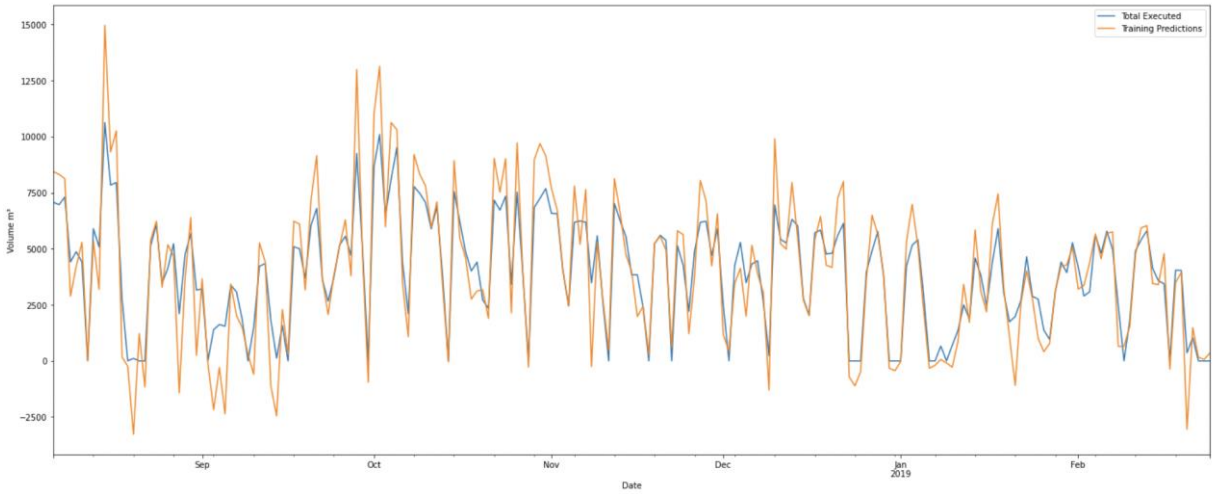


Figure 35 ARIMA model actual and predicted values in the train set

The walk forward validation method was performed on the selected model. For the validation set, the model produced an RMSE of 1100 m^2 and an R^2 of 0.8. The model is slightly overfitting since the RMSE of the validation set is a bit lower than the RMSE of the training set. Various attempts were performed to enhance the model and to remove overfitting issues were unsuccessful. Figure 36 shows the predicted and actual values of the production values for the validation set using the SARIMAX model.

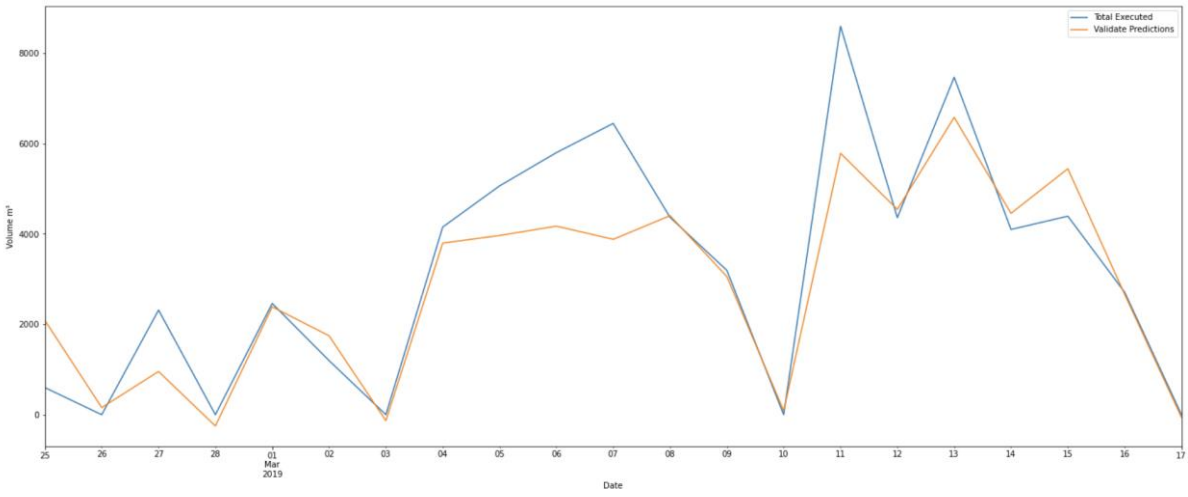


Figure 36 ARIMA model actual and predicted values in the validation set

iii. Diagnosis

Moving to the “diagnosis” step, the residual plots of the training sample were analyzed to check the goodness of fit of the model and to check that all assumptions are satisfied. Figure 37 presents the diagnosis plots that include the residual plot, histogram, normal Q-Q plot and the ACF correlogram of the residuals. The ACF correlogram plot of the residual shows that the residuals are not correlated since the ACF values are insignificant. Also, the residuals have a normal distribution with a mean close to zero, and the variance is constant. Thereby, the model is considered to be a good fit.

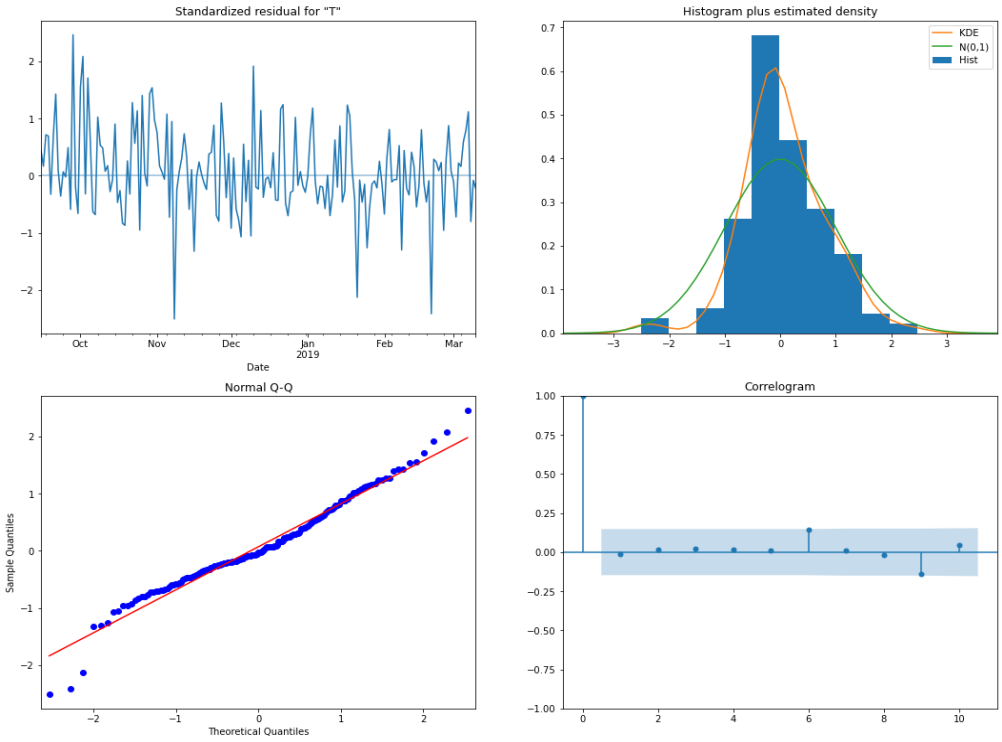


Figure 37 ARIMA model residual diagnostics

Once the residuals are verified, the selected model was evaluated on the test set. The test set provided an RMSE 914 m^2 of and an R^2 of 0.74. Figure 38 shows the

predicted and actual values of the production values for the validation set using the SARIMAX model.

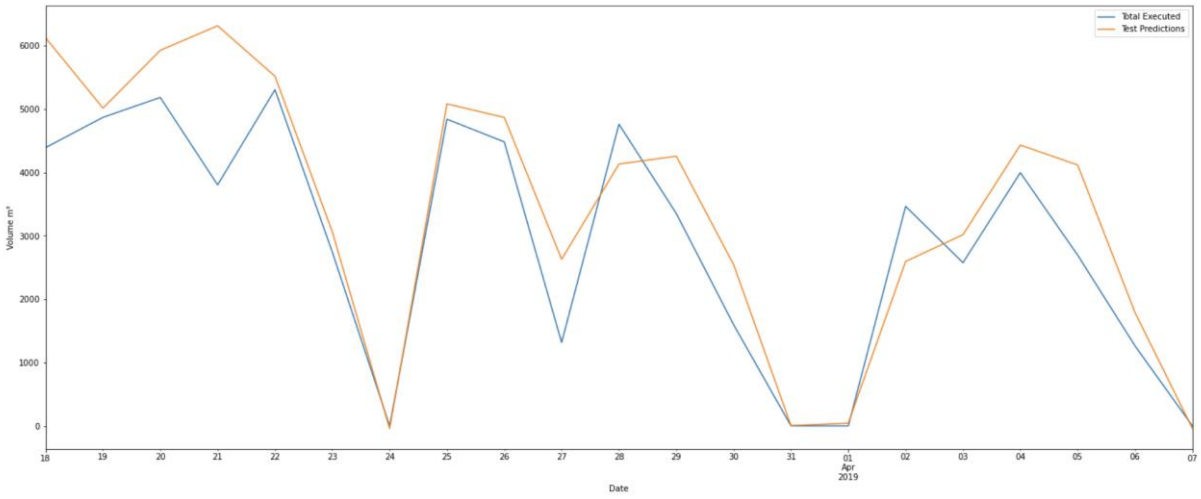


Figure 38 ARIMA model actual and predicted values in the test set

Table 19 ARIMA model error metrics summary results

	MSE	RMSE	R^2
Training Set	1710395	1308	0.71
Validation Set	1189314	1100	0.80
Test Set	834551	914	0.74

2. Benchmark Model of MLP with Random Shuffle and no Time Lag

The MLP ANN was developed in Python using the Keras library. When developing the model, various potential models were tested and evaluated, but several ones failed to meet the required evaluation conditions of training, validation, stability, and testing requirements. Consequently, several other models were tested by modifying the various hyperparameters in a systematic way. A suggested model is presented in the following sections.

In this model, the data split was performed randomly and not in a sequential manner. This model serves as a benchmark to check the validity of incorporating the time lag in an ANN model to predict production output.

It uses as input the exogenous factors that are the weather features (3) and resource hours (7) of the corresponding day. The total number of input features becomes 10.

The network architecture of the MLP model is comprised of one input layer, one hidden layer, and one output layer. In the input layer, the number of neurons is equivalent to the number of input features (10). In the hidden layer, 30 neurons were used. For the output layer, only one neuron is designated since it is a regressions problem where the production value is being forecasted.

The RELU activation function was used in the hidden layer due to its superior performance. Also, the Adam optimizer was used for the training algorithm since it proves a superior performance in the literature and in the model performance. The default values were used for the parameters $\beta 1$ (0.9) and $\beta 2$ (0.999). A learning rate η of 0.001 was selected. 115 number of training epochs was selected for building the model.

The MSE and RMSE function was adopted to evaluate the performance of the model. To ensure that the model is not prone to overfitting problems, the Ridge ($\ell 2$) and Lasso ($\ell 1$) regularization methods were adopted. The selected Ridge and Lasso hyperparameters were respectively 0.000001 and 0.000001 for the MLP model.

The following hyperparameters were being updated between every model iteration: the number of lags with the corresponding number of input features,

number of neurons in the input layer and each hidden layer, number of hidden layers, regularization method, number of training epoch, and the learning rate.

Table 20 below summarizes the selected hyperparameters for the MLP model without a time lag.

Table 20 MLP with no time lag model hyperparameters and characteristics

	MLP model
Network architecture:	
Number of hidden layers	1
Number of neurons in input layer	10
Number of neurons in each hidden layer	30
Number of neurons in output layer	1
Network activation function:	
Activation function in hidden layer (s)	ReLu
Activation function in output layer	Linear
Optimization algorithm:	Adam
Momentum decay hyperparameter: $\beta 1$	0.9
Scaling decay hyperparameter: $\beta 2$	0.999
Learning rate: η	0.001
Training batch size	
Number of training epoch	115
Performance function	MSE
Regularization:	
Ridge regularization $\ell 1$	0.000001
Lasso regularization $\ell 2$	0.000001

The selected hyperparameters were used to fit and evaluate the model. Table 21 summarizes the results of the MLP model without a time lag.

Table 21 MLP with no time lag model error metrics summary results

	MSE	**RMSE**	**R^2**
Training Set	1664288	1290	0.7
Validation Set	2360119	1536	0.72
Test Set	1972197	1404	0.69

The model has a RMSE of 1290 m^3 on the training set and an R^2 of 0.7. The model can explain 70% of the training set. Figure 39 shows the predicted and actual values of the production values for the training set using the MLP model without a time lag.

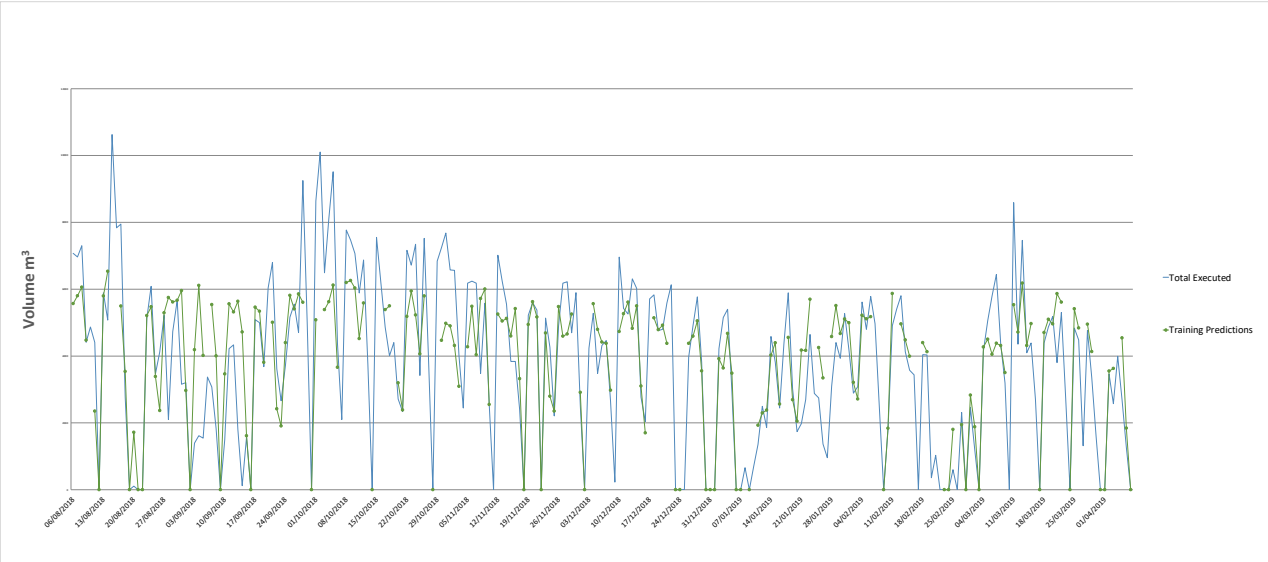


Figure 39 MLP with no time lag model actual and predicted values in the train set

The model was further evaluated on the validation set. A RMSE of 1,536 m^3 was obtained along with a R^2 of 0.72. The validation RMSE is approximately close to the RMSE of the training set; thus, the model is not overfitting. Also, the fact that the validation error is a bit higher than the training error signifies that the model is valid because the model is anticipated to underperform on the data which was not used for fitting. In addition, the model is explaining 72% of the data. Figure 40 shows the predicted and the actual values of the production for the validation set using the MLP model without a time lag.

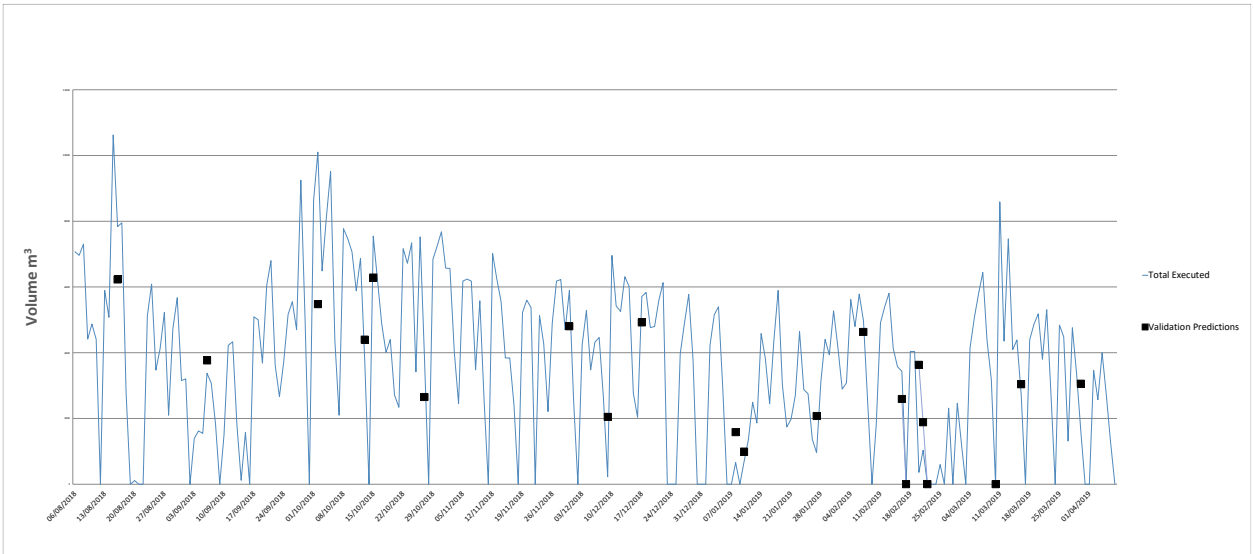


Figure 40 MLP with no time lag model actual and predicted values in the validation set

Due to the stochastic nature of the MLP models, the model was fit 30 times using the same hyperparameters to ensure its stability. Through every iteration, the initial weights is the only parameter that is changing because of the stochastic nature of the neural network optimization algorithm. The mean average for the train RMSE is 1,301 m^3 with a standard deviation of 32. Also, the mean average of the validation RMSE was 1,604 m^3 and a standard deviation of 63. The model is considered to be stable since the average train and validation errors are close to the selected model's corresponding train and validation errors. Also, both have a low standard deviation. Thereby, the model's hyperparameters show adequate results.

The model was further evaluated on the test set using the walk-forward validation. A test set RMSE of 1,404 m^3 was obtained along with a R^2 of 0.69. The error is slightly higher than the train RMSE; hence, the model is not overfitting. The results show an adequate fit since the model is explaining 69% of the data. Figure 41 shows the predicted and the actual values of the production for the test set using the MLP model without a time lag.

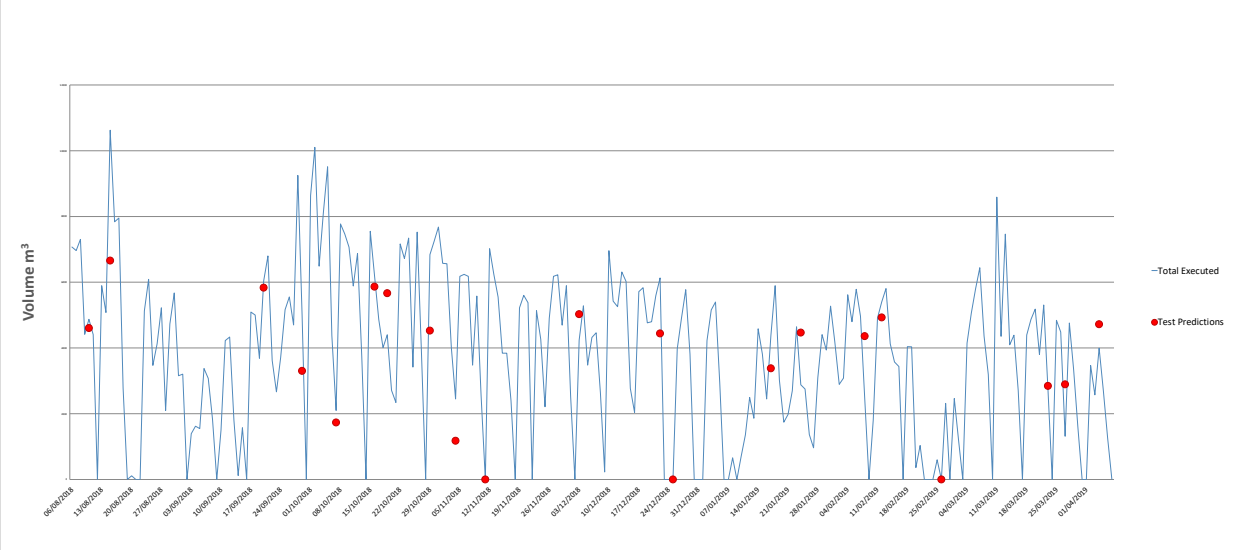


Figure 41 MLP with no time lag model actual and predicted values in the test set

D. Model Comparison and Discussion

Table 22 summarizes the obtained model fit RMSEs and R^2 on the train, validation, and test sets of the SARIMAX, MLP time series model, LSTM, GRU, MLP model without a time lag models.

Table 22 Summary Results of the developed models

Type of model	MSE train	RMSE train	R^2 train	MSE validate	RMSE validate	R^2 vaildate	MSE test	RMSE test	R^2 test
SARIMAX	1710395	1308	0.71	1189314	1100	0.80	834551	914	0.74
MLP- no time lag	1664288	1290	0.7	2360119	1536	0.72	1972197	1404	0.69
MLP time series	662596	814	0.89	1052676	1026	0.85	687536	829	0.81
LSTM	723706	850	0.87	1406069	1185	0.78	756878	869	0.77
GRU	791897	890	0.87	1463751	1209	0.78	818536	904	0.75

1. Comparison of the different ANN Time Series Models

The time series models produced significant results. The MLP time series model produced more satisfactory results compared with other ANN time series models. The training set was able to fit the data with a low error and a significant R^2 fit. The MLP time series model is able to explain 89% of training data set which is a significant value. Also, the validation and test set errors are considered to be low given the context of the problem and given the range of the data from 0 to 10,000 m^3.

The LSTM and GRU produced very similar results. This was anticipated since both models behave in a similar way. They were both able to explain 87% of the training set. Their performance is satisfactory, but the MLP time series model produced a better fit for this case study in all 3 sets.

2. Comparison of ANN Time Series Models with SARIMAX Model

The R^2 of the MLP time series across all the sets is higher than the R^2 of the statistical time series SARIMAX model. Also, the RMSE of the MLP time series across all sets is lower than the RMSE of the SARIMAX model, thereby the MLP time series has a better ability in generalizing to unforeseen and new data for this case study. In this case study, the SARIMAX model overfitted the data. One of the downturns of using a statistical model is the limitation in optimizing the model's parameters because the parameters to be tuned are few. Whereas the neural network models are more flexible for optimization since they contain various hyperparameters to be tuned. Also, the statistical model's properties limit its accuracy because it considers that the time series is produced from linear processes, so it might be inappropriate for real-world problems that are

nonlinear (Khashei et al., 2009). Nonlinear and time-variant problems need to be tackled with tools that can model the complex behavior that changes with time.

For this case study, the MLP time series model showed satisfactory results and produced an improvement over the traditional state-of-the-art ARIMA model. Although the statistical time series model SARIMAX was able to explain 70-80% of the data, the MLP time lag model outperformed it across all sets. Thus, the MLP time series act as a beneficial tool in forecasting the field production value of the upcoming week in a construction activity.

Table 23 Error metric comparison of ANN time series to SARIMAX

	RMSE difference	**RMSE improvement**	**R^2 difference**
Training Set	494	37%	0.18
Validation Set	74	7%	0.05
Test Set	85	10%	0.07

3. Comparison of ANN Time Series Models with MLP- no Time Lag Model

It is evident that incorporating the time factor as a predictor improves the accuracy of the predictive model. All three ANN time series models (MLP time series, LSTM, and GRU), outperformed the MLP model without a time lag. So, this shows the importance of including the time lag as a predictor to account for the time dependency and sequence in the data. Hence, patterns are detected during the performance of an activity. The MLP model without a time lag can only explain 70% of the training set compared with a value of 89% for the MLP time series model. Comparing the performance of the MLP time series model to the performance of the MLP without a time lag model, Table 24 summarizes the comparison of the metrics between both models. Across all three sets, it

is significant that there is approximately 30-40% improvement in the RMSE error metric and a 12-19% improvement in the R^2 fit of the data.

Table 24 Error metric comparison of ANN time series to MLP model with no time lag

	RMSE difference	RMSE improvement	R^2 difference
Training Set	476	37%	0.19
Validation Set	510	33%	0.13
Test Set	575	41%	0.12

4. Comparison of the Models with the Project Manager's Approach

As previously mentioned, the PM of this project was forecasting the excavation production on a weekly basis by relying on the traditional approach of relying on subjective judgment and historical data. However, the subjective judgment leads to uncertainty, and relying on the historical data disregards the fact that every project is unique and changes with time. The RMSE and R^2 were calculated for the PM traditional method of forecasting values compared with the actual values. Table 25 summarizes the error metrics of the PM forecast across the train, validation, and test set.

Table 25 Error metrics PM forecasts

	MSE	RMSE	R^2
Training Set	4864187	2205	0.41
Validation Set	6506320	2551	0.56
Test Set	2196178	1482	0.45

The RMSE of the training set is 2205 m^2 which is a very high value considering the data at hand. The PM's forecast is only able to explain 41%, which is a low value.

Also, the validation and test set RMSE have a high value, signifying that the PM is unable to properly forecast the production value.

The RMSE of the PM traditional forecasting method is 3 times higher than the RMSE of the MLP time series model, which is the best performing model in the case study. Table 26 summarizes the comparison in the performance metrics between the PM traditional forecasting method and the MLP time series model, which is the best performing model in this case study. There is a 44-60% improvement in the RMSE across all 3 sets; in addition, the data has a better fit due to the high R^2 difference that ranges from 36-48%.

Table 26 Error metrics comparison of ANN time series to PM forecasts

	RMSE difference	**RMSE improvement**	**R^2 difference**
Training Set	1391	63%	0.48
Validation Set	1525	60%	0.29
Test Set	653	44%	0.36

Consequently, the MLP time series model can considerably improve the forecasting accuracy compared with the PM's adopted traditional forecasting method.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

Construction productivity fluctuates along the construction phase of a project due to various operational, managerial, and environmental factors. When it comes to performance prediction during construction, the traditional ways of relying on subjective judgment and on historical data fail to account for the dynamic nature of construction activities. Despite previous efforts to develop data-driven productivity predictive models, few studies have sufficiently employed advanced tools to integrate the time factor in predicting performance of ongoing activities. It is evident that regular and frequent prediction over time is required to effectively manage construction resources along the execution phase. As such, this paper presented a generic data-driven framework to accurately forecast ongoing field production of construction operations using ANN time series.

The methodology of this research work was divided into two parts. The first part presented a generic framework that assists project managers in the execution of field work. The framework includes a Machine Learning time series model which uses field production information to accurately forecast future the production output. The model includes various neural networks (MLP, LSTM and GRU) that can forecast production values while considering the time factor in addition to other external factors. The model also allows for establishing a resource management plan by estimating the resources needed to achieve the corresponding performance.

The second part the research work included developing two benchmark models: 1) a statistical based time series model to analyze and forecast field production based on

internal and external factors, and 2) an MLP model that does not incorporate the time factor (i.e., time lag). The results in this part were compared with the results of the ML time series model.

Upon testing the methodology on a case study of a large infrastructure project over the period of eight months, the developed time series based methodology showed a satisfactory performance. The MLP time series model outperformed both the ANN time series models (GRU and LSTM) and the statistical ARIMA model. The MLP time series model was able to produce an R^2 fit of 0.89 with a RMSE of 814 m^3, while the ARIMA model produced an R^2 fit of 0.71 with a RMSE of 1308 m^3. In addition, the MLP time series model provided a better fit compared with the MLP model without a time lag that produced an R^2 fit of 0.7 with a RMSE of 1,290 m^3. Thus, incorporating time as a predictor can improve the model's accuracy. Also, the developed model's performance was compared with the forecasting strategy that was adopted on the project. The MLP time series model provided a better fit compared with the forecasting strategy of the project. The forecasting accuracy of the strategy adopted on the project was low, producing an R^2 fit of 0.48 with a RMSE of 1391 m^3.

This study enriches the literature by developing a data-driven framework that is able to collect and predict field production for a construction activity using advanced ANN time series by extracting useful insights from ongoing field data. The adopted ANN time series predictive model can model the non-linear changing behavior of an activity over time, thereby producing reliable forecasts. Since the framework integrates the relevant ongoing field data, the dynamic behavior of the project is captured over time.

In practice, the developed framework aims to assist project managers in improving their production forecast accuracy of the upcoming days, thereby reducing

uncertainty in anticipating the fluctuating productivity. Also, this tool can help project managers in proactively planning their resources ahead of time to mitigate any unforeseen performance fluctuation. Extracting useful insights from ongoing field data can improve the decision making of project managers. Therefore, project managers should be equipped with the required practical knowledge in data analytics to successfully apply this framework.

A limitation of this study is that it is performed on repetitive and continuous activities spanning a significant duration along the project in order to have sufficient data points for training. To account for project specific parameters and data, the PM can start using the model a few weeks after the project kickoff. The model could be used earlier, i.e. in the planning phase, using historical data from similar projects and forecasted site conditions, but this may achieve less accuracy. Another limitation is the fact that the resource hours and the weather factors used for forecasting are the real values instead of the estimated ones. Due to the lack of access to such information, the real data was used. This framework should be further tested on different type of activities to validate its accuracy. In this study, only the weather factors and the resource hours are considered as external factors.

Future work may entail including more factors related to the corresponding activity to improve the performance. In addition, this study could be extended by integrating an optimized algorithm model to output the resource hours required by inputting the desired production output.

REFERENCES

Abdelhamid, T. S., & Everett, J. G. (1999). Time series analysis for construction productivity experiments. *Journal of construction engineering and management*, *125*(2), 87-95.

AbouRizk, S., Knowles, P., & Hermann, U. R. (2001). Estimating labor production rates for industrial construction activities. *Journal of construction engineering and management*, *127*(6), 502-511.

Alaghbari, W., Al-Sakkaf, A. A., & Sultan, B. (2019). Factors affecting construction labour productivity in Yemen.*International Journal of Construction Management*, *19*(1), 79-91.

Battiti, R. (1992). First-and second-order methods for learning: between steepest descent and Newton's method. *Neural computation*, *4*(2), 141-166.

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). Recurrent neural networks for short-term load forecasting: an overview and comparative analysis: Springer.

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2019). Single and Multi-Sequence Deep Learning Models for Short and Medium Term Electric Load Forecasting. *Energies*, *12*(1), 149. MDPI AG. Retrieved from http://dx.doi.org/10.3390/en12010149

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

Brockwell, P. J., Brockwell, P. J., Davis, R. A., & Davis, R. A. (2016). *Introduction to time series and forecasting*. springer.

Chao, L. C., & Skibniewski, M. J. (1994). Estimating construction productivity: Neural-network-based approach.*Journal of Computing in Civil Engineering*, *8*(2), 234-251.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, J., & Ryu, H. G. (2015). Statistical analysis of construction productivity for highway pavement operations.*KSCE Journal of Civil Engineering*, *19*(5), 1193-1202.

Choy, E., & Ruwanpura, J. Y. (2005). Situation based modeling for construction productivity. In *Construction Research Congress 2005: Broadening Perspectives* (pp. 1-10).

Claveria, O., Monte, E., & Torra, S. (2015). Tourism demand forecasting with neural network models: different ways of treating information. *International Journal of Tourism Research*,*17*(5), 492-500.

Demuth, H., Beale, M., & Hagan, M. (2009). Neural Network Toolbox 6 User's Guide'The MathWorks. *Inc, MA, US*.

Deshmukh, D. A., & Mahatme, P. S. (2016). Factors Affecting Performance of Excavating Equipment: An Overview.*International Journal of Science and Research (IJSR) ISSN (Online)*, 2319-7064.

Díaz-Robles, L. A., Ortega, J. C., Fu, J. S., Reed, G. D., Chow, J. C., Watson, J. G., & Moncada-Herrera, J. A. (2008). A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco, Chile. *Atmospheric Environment*, *42*(35), 8331-8340.

Dozzi, S. P., & AbouRizk, S. M. (1993). *Productivity in construction*. Ottawa: Institute for Research in Construction, National Research Council.

Edwards, D. J., & Griffiths, I. J. (2000). Artificial intelligence approach to calculation of hydraulic excavator cycle time and output. *Mining Technology*, *109*(1), 23-29.

Edwards, D. J., & Holt, G. D. (2000). ESTIVATE: a model for calculating excavator productivity and output costs.*Engineering, Construction and Architectural Management*.

El-Gohary, K. M., Aziz, R. F., & Abdel-Khalek, H. A. (2017). Engineering approach using ANN to improve and predict construction labor productivity under different influences.*Journal of Construction Engineering and Management*, *143*(8), 04017045.

Everett, J. G., & Farghal, S. (1994). Learning curve predictors for construction field operations. *Journal of construction engineering and management*, *120*(3), 603-616.

Ezeldin, A. S., & Sharara, L. M. (2006). Neural networks for estimating the productivity of concreting activities. *Journal of construction engineering and management*, *132*(6), 650-656.

Farghal, S. H., & Everett, J. G. (1997). Learning curves: Accuracy in predicting future performance. *Journal of construction engineering and management*, *123*(1), 41-45.

Gates, M., & Scarpa, A. (1972). Learning and experience curves. *Journal of the Construction Division*, *98*(1), 79-101.

Gavin, H. (2011). The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, *28*, 1-5.

Gerek, I. H., Erdis, E., Mistikoglu, G., & Usmen, M. (2015). Modelling masonry crew productivity using two artificial neural network techniques. *Journal of Civil Engineering and Management*, *21*(2), 231-238.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and Tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Ghoddousi, P., & Hosseini, M. R. (2012). A survey of the factors affecting the productivity of construction projects in Iran. *Technological and economic development of economy*,*18*(1), 99-116.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In*Proceedings of the thirteenth international*

conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.

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

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey.*IEEE transactions on neural networks and learning systems*,*28*(10), 2222-2232.

Hall,M.A.1998."Correlation based feature selection for machine learning ." Ph.D. thesis, Univ. o fWaikato

Halpin, D. W., and Riggs, L. S. (1997). *Planning and analysis of construction operations*, Wiley, New York.

Hanna, A. S., & Gunduz, M. (2004). Impact of change orders on small labor-intensive projects. *Journal of Construction Engineering and Management*, *130*(5), 726-733.

Hanna, A. S., Taylor, C. S., & Sullivan, K. T. (2005). Impact of extended overtime on construction labor productivity. *Journal of construction engineering and management*, *131*(6), 734-739.

Hasan, A., Baroudi, B., Elmualim, A., & Rameezdeen, R. (2018). Factors affecting construction productivity: a 30 year systematic review. *Engineering, Construction and Architectural Management*.

Heravi, G., & Eslamdoost, E. (2015). Applying artificial neural networks for measuring and predicting construction-labor productivity. *Journal of Construction Engineering and Management*, *141*(10), 04015032.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hola, B., & Schabowicz, K. (2010). Estimation of earthworks execution time cost by means of artificial neural networks.*Automation in Construction*, *19*(5), 570-579.

Hwang, B. G., Zhu, L., & Ming, J. T. T. (2016). Factors affecting productivity in green building construction projects: The case of Singapore. *Journal of Management in Engineering*,*33*(3), 04016052.

Hwang, S. (2010). Cross-validation of short-term productivity forecasting methodologies. *Journal of construction engineering and management*, *136*(9), 1037-1046.

Hwang, S., & Liu, L. Y. (2010). Contemporaneous time series and forecasting methodologies for predicting short-term productivity. *Journal of Construction Engineering and Management*, *136*(9), 1047-1055.

Hyndman, R. (2014). Variations on Rolling Forecasts. Retrieved April 28, 2022, from https://robjhyndman.com/hyndsight/rolling-forecasts/

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

Jabri, A., & Zayed, T. (2017). Agent-based modeling and simulation of earthmoving operations. *Automation in Construction*, *81*, 210-223.

Jarkas, A. M. (2010). Critical investigation into the applicability of the learning curve theory to rebar fixing labor productivity.*Journal of Construction Engineering and Management*,*136*(12), 1279-1288.

Kassem, M., Mahamedi, E., Rogage, K., Duffy, K., & Huntingdon, J. (2021). Measuring and benchmarking the productivity of excavators in infrastructure projects: A deep neural network approach. *Automation in Construction*, *124*, 103532.

Khashei, M., Bijari, M., & Ardali, G. A. R. (2009). Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). *Neurocomputing*, *72*(4-6), 956-967.

Kingman, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. Conference paper. In *3rd International Conference for Learning Representations*.

Kulkarni, P., Londhe, S., & Deo, M. (2017). Artificial neural networks for construction management: a review. *Journal of Soft Computing in Civil Engineering*, *1*(2), 70-88.

Laptev, N., Yosinski, J., Li, L. E., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at uber. In *International conference on machine learning* (Vol. 34, pp. 1-5).

Mawdesley, M. J., & Al-Jibouri, S. (2009). Modelling construction project productivity using systems dynamics approach. *International Journal of Productivity and Performance Management*.

Mirahadi, F., & Zayed, T. (2016). Simulation-based construction productivity forecast using neural-network-driven fuzzy reasoning. *Automation in Construction*, *65*, 102-115.

Moselhi, O., Assem, I., & El-Rayes, K. (2005). Change orders impact on labor productivity. *Journal of Construction Engineering and Management*, *131*(3), 354-359.

Moselhi, O., Hegazy, T., & Fazio, P. (1991). Neural networks as tools in construction. *Journal of construction engineering and management*, *117*(4), 606-625.

Motwani, J., Kumar, A., and Novakoski, M. 1995. "Measuring construction productivity: A practical approach." Int. J. Prod. Perform. Manage., 448, 18–20.

Naoum, S. G. (2016). Factors influencing labor productivity on construction sites. *International Journal of Productivity and Performance Management*.

Nasirzadeh, F., & Nojedehi, P. (2013). Dynamic modeling of labor productivity in construction projects. *International journal of project management*, *31*(6), 903-911.

Nau, R. (2016). Statistical Forecasting: Notes on Regression and Time Series Analysis. Fuqua School of Business, Duke University, Durham.

Nojedehi, P., & Nasirzadeh, F. (2017). A hybrid simulation approach to model and improve construction labor productivity.*KSCE Journal of Civil Engineering*, *21*(5), 1516-1524.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Ok, S. C., & Sinha, S. K. (2006). Construction equipment productivity estimation using artificial neural network model.*Construction Management and Economics*, *24*(10), 1029-1044.

Oral, E. L., & Oral, M. (2010). Predicting construction crew productivity by using Self Organizing Maps. *Automation in Construction*, *19*(6), 791-797.

Oral, M., Oral, E. L., & Aydın, A. (2012). Supervised vs. unsupervised learning for construction crew productivity prediction. *Automation in Construction*, *22*, 271-276.

Park, H. S. (2006). Conceptual framework of construction productivity estimation. *KSCE Journal of Civil Engineering*,*10*(5), 311-317.

Petneházi, G. (2019). Recurrent neural networks for time series forecasting. *arXiv preprint arXiv:1901.00069*.

Plummer, E. (2000). Time series forecasting with feed-forward neural networks: guidelines and limitations. *Neural Networks*,*1*, 1.

Pocock, J. B., Hyun, C. T., Liu, L. Y., & Kim, M. K. (1996). Relationship between project interaction and performance indicators. *Journal of construction engineering and management*, *122*(2), 165-176.

Polat, G., & Arditi, D. (2005). The JIT materials management system in developing countries. *Construction Management and Economics*, *23*(7), 697-712.

Portas, J., & AbouRizk, S. (1997). Neural network model for estimating construction productivity. *Journal of construction engineering and management*, *123*(4), 399-410.

Rao, G. N., and Grobler, F. (1997). "Integrated analysis of cost risk and schedule risk." *Proc., 4th Congress on Computing in Civil Engineering*, ASCE, New York, 1404–1411.

Rashidi, A., Nejad, H. R., & Maghiar, M. (2014). Productivity estimation of bulldozers using generalized linear mixed models. *KSCE journal of civil engineering*, *18*(6), 1580-1589.

Russell, J. S., Jaselskis, E. J., & Lawrence, S. P. (1997). Continuous assessment of project performance. *Journal of construction engineering and management*, *123*(1), 64-71.

Sanders, S. R., & Thomas, H. R. (1991). Factors affecting masonry-labor productivity. *Journal of Construction Engineering and Management*, *117*(4), 626-644.

Schäfer, A. M., & Zimmermann, H. G. (2007). Recurrent neural networks are universal approximators. In *International Conference on Artificial Neural Networks* (pp. 632-640). Springer, Berlin, Heidelberg.

Seresht, N. G., & Fayek, A. R. (2020). Neuro-fuzzy system dynamics technique for modeling construction systems.*Applied Soft Computing*, *93*, 106400.

Seresht, N.G, & Fayek, A. R. (2018). Dynamic modeling of multifactor construction productivity for equipment-intensive activities. *Journal of Construction Engineering and Management*, *144*(9), 04018091.

Shaheen, A. A., Fayek, A. R., & AbouRizk, S. M. (2009). Methodology for integrating fuzzy expert systems and discrete event simulation in construction engineering. *Canadian Journal of Civil Engineering*, *36*(9), 1478-1490.

Shehata, M. E., & El-Gohary, K. M. (2011). Towards improving construction labor productivity and projects' performance.*Alexandria Engineering Journal*, *50*(4), 321-330.

Shi, J. J. (1999). A neural network based system for predicting earthmoving production. *Construction Management & Economics*, *17*(4), 463-471.

Shi, J., & AbouRizk, S. S. (1998). An automated modeling system for simulating earthmoving operations. *Computer‐Aided Civil and Infrastructure Engineering*, *13*(2), 121-130.

Shmueli, G., & Lichtendahl Jr, K. C. (2016). Practical time series forecasting with r: A hands-on guide. Axelrod Schnall Publishers.

Song, L., & AbouRizk, S. M. (2008). Measuring and modeling labor productivity using historical data. *Journal of construction engineering and management*, *134*(10), 786-794.

Sonmez, R., & Rowings, J. E. (1998). Construction labor productivity modeling with neural networks. *Journal of construction engineering and management*, *124*(6), 498-504.

Sparkes, J. R., & McHugh, A. K. (1984). Awareness and use of forecasting techniques in British industry. *Journal of Forecasting*, *3*(1), 37-42.

Tabachnick, B. G., & Fidell, L. S. (2018). Using Multivariate Statistics, 7th Edn London.

Talhouni, B. T. (1990). *Measurement and analysis of construction labour productivity* (Doctoral dissertation, University of Dundee).

Tam, C. M., Tong, T. K., & Tse, S. L. (2002). Artificial neural networks model for predicting excavator productivity.*Engineering, construction and architectural management*.

Teixeira, J. P., & Fernandes, P. O. (2012). Tourism time series forecast-different ANN architectures with time index input. *Procedia Technology*, *5*, 445-454.

Thomas, H. R., & Raynar, K. A. (1997). Scheduled overtime and labor productivity: quantitative analysis. *Journal of construction engineering and management*, *123*(2), 181-188.

Tsehayae, A. A., & Fayek, A.R. (2014). Identification and comparative analysis of key parameters influencing construction labour productivity in building and industrial projects. *Canadian Journal of Civil Engineering*, *41*(10), 878-891.

Yang, J., Edwards, D. J., & Love, P. E. (2003). A computational intelligent fuzzy model approach for excavator cycle time simulation. *Automation in construction*, *12*(6), 725-735.

Yunpeng, L., Di, H., Junpeng, B., & Yong, Q. (2017, November). Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In *2017 14th web information systems and applications conference (WISA)* (pp. 305-310). IEEE.

Zayed, T. M., & Halpin, D. W. (2005). Productivity and cost regression models for pile construction. *Journal of Construction Engineering and Management*, *131*(7), 779-789.