APPLICATION OF ARTIFICIAL INTELLIGENCE TO WASTEWATER TREATMENT PLANT OPERATION

by

Praewa Wongburi

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Civil and Environmental Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2021

Date of final oral examination: 07/28/2021

The dissertation is approved by the following members of the Final Oral Committee:

Jae Park. Faculty, Professor, Civil and Environmental Engineering

Greg Harrington. Faculty, Professor, Civil and Environmental Engineering

Chin Wu. Faculty, Professor, Civil and Environmental Engineering

Andrea Hicks. Faculty, Assistant Professor, Civil and Environmental Engineering

Miaoyan Wang. Faculty, Assistant Professor, Statistics

ABSTRACT

In a wastewater treatment plant (WWTP), big data is collected from sensors installed in various unit processes, but limited data is used for operation and regulatory permit requirements. With the advancement in information technology, the data size in wastewater treatment systems has increased significantly. However, WWTPs have not used big data systematically to aid the operation and detect potential operational issues due to the lack of specialized analytical tools.

The objectives of the study were to: (1) develop analytics methods suitable for the management of big data generated in WWTPs, (2) interpret analytics results for extracting meaningful information, (3) implement a recurrent neural network (RNN) and Long Short-Term Memory (LSTM) to predict effluent water quality parameters and Sludge Volume Index (SVI), (4) apply an Explainable Artificial Intelligence (AI) algorithm to determine causes of predicted values, and (5) propose a real-time control using a predictive model to monitor and optimize the operation of WWTPs.

The predictive AI models in WWTPs were developed by applying big data analytics, statistical analysis, and RNN algorithms with an Explainable AI algorithm. The models successfully and accurately predicted the effluent water quality data and a key operational parameter, SVI. Furthermore, the Explainable AI algorithm provided insight into which influent parameters affected higher predicted effluent concentrations and SVI on a specific day, allowing operators to take corrective actions.

From a WWTP's operational data analysis, the RNN model successfully predicted the effluent concentrations of BOD_5, total nitrogen (TN) and total phosphorus (TP), and SVI. Furthermore, the

Explainable AI analysis found that higher influent NH_3N values lead to higher effluent BOD_5, and higher influent total suspended solids (TSS) and TP values resulted in lower effluent BOD_5, implying the importance of controlling dissolved oxygen (DO) in aeration basins. Since aeration is one of the major energy consumption sources in WWTPs, real-time prediction of the effluent water quality using the self-learning AI system developed in this study can be adopted to lower the energy cost significantly while improving effluent water quality. WWTPs must develop control methods based on the RNN prediction and Explainable AI analysis due to different operational conditions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background and Research Motivation

Due to the demands for lower operation and maintenance cost, energy cost, stringent compliance requirements of water quality parameters, and lower greenhouse gas emission, WWTPs (WWTPs) must be in a smart management mode. Due to the extensive use of water quality sensors, big data is generated every day or even every second. In the wastewater treatment sector and many other fields such as finance, marketing, stocks, health care, and so on, big data plays an essential role. Data generation has been tremendously increasing since 2010, and 90% of the world's data has been created in the past two years (Figure 1.1).

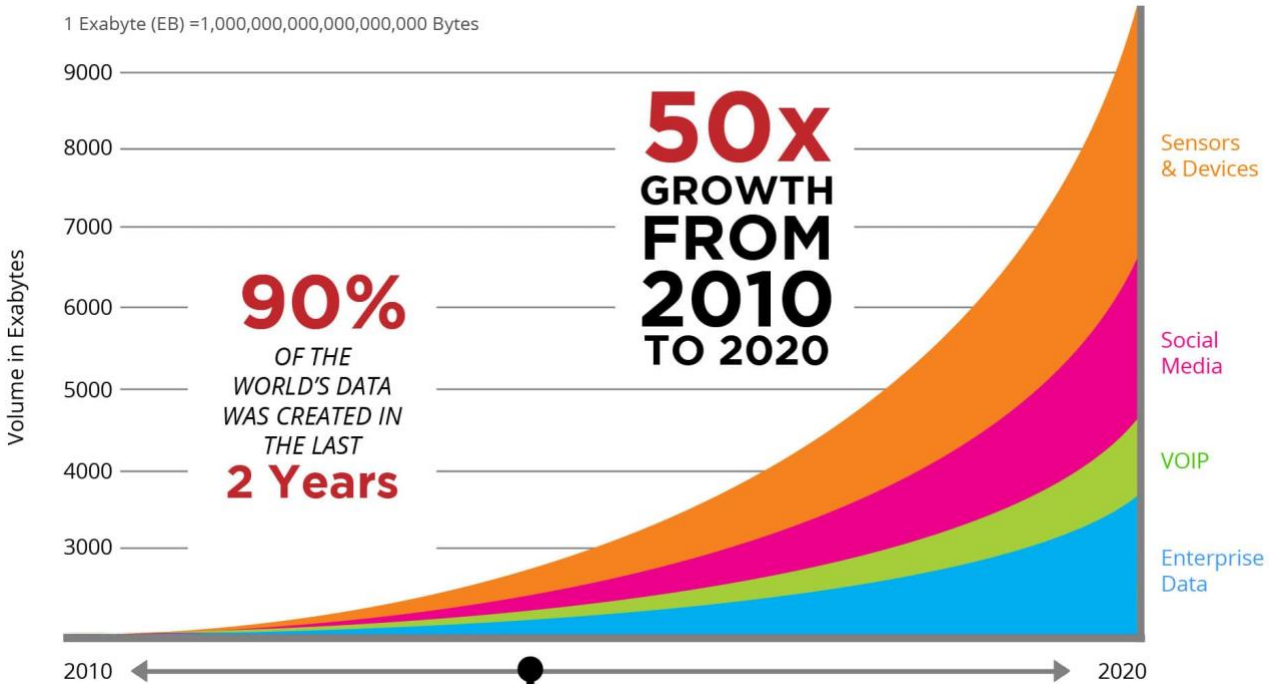


Figure 1.1 The growth of data from 2010 to 2020. (Source: Roser et al., 2015)

Big data is currently most preferably exploited in organizations, companies, and businesses. However, a massive amount of data results in the difficulties of storing, monitoring, analyzing, and visualizing data for further processing. Therefore, data is often stored and then underutilized.

A good understanding of the data dynamics in WWTPs is vital for reliable monitoring and control activities. However, the dynamical behavior of the data is usually complicated and uncertain due to nonlinearity, variations from the environmental conditions, strong interactions between the process variables involved, and changes in the flow rate and concentration of the composition of the influent (Harrou et al., 2018). Finding insight from historical and real-time data can directly improve traditional operational systems in a better direction.

A series of wastewater treatment processes remove pollutants from wastewater to be safely reused or discharged into natural water resources. Treated water can be recycled and redistributed for agricultural, industrial, and other purposes or safely released back into the natural resources without causing any adverse effects (Grant et al., 2012). The effluent from a WWTP must meet the National Pollutant Discharge Elimination System (NPEDES) permit to protect the environment and public health (Siegrist, 2017). Lack of access to safe water leads to a risk factor for infectious diseases such as cholera, diarrhea, and dysentery. According to the Global Burden of Disease study, 1.2 people died prematurely in 2017 due to unsafe water (Figure1.2). This number was three times the number of homicides in 2017 and approximately equal to the amount that died in road accidents globally. Besides, only 71% of the world population has access to safe drinking water, which means that 29% of the world population does not have access to safe water. It equates to 2.1 billion people globally (Figure 1.3).

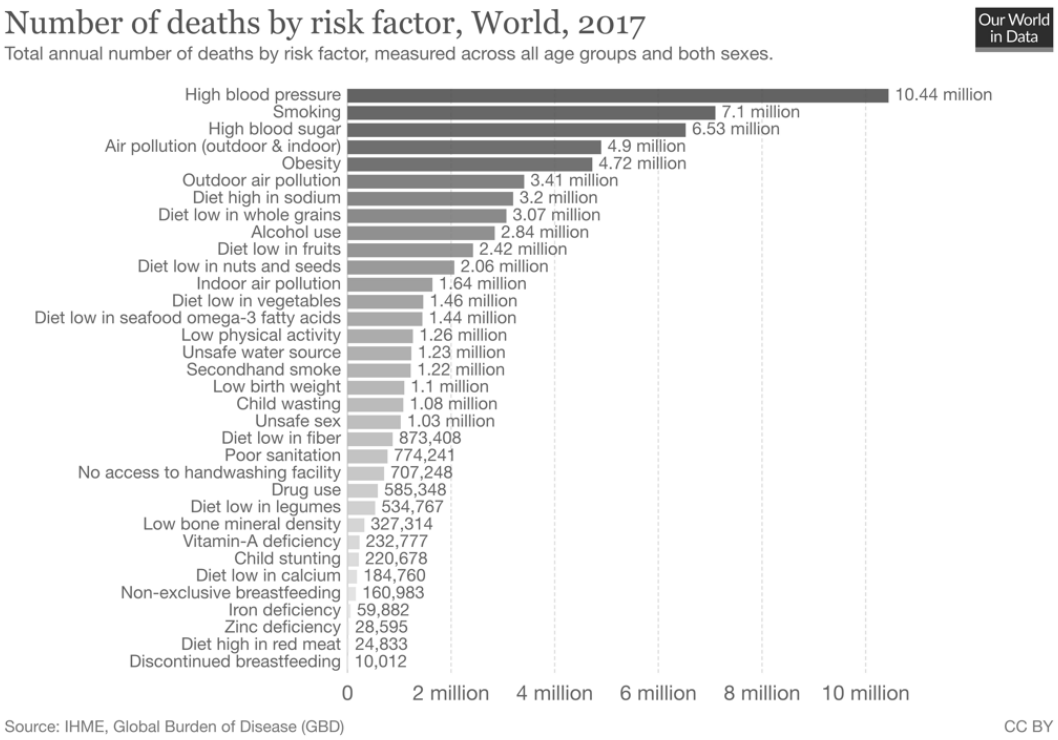


Figure 1.2 Number of deaths by risk factor. (Source: Ritchie & Roser, 2018)

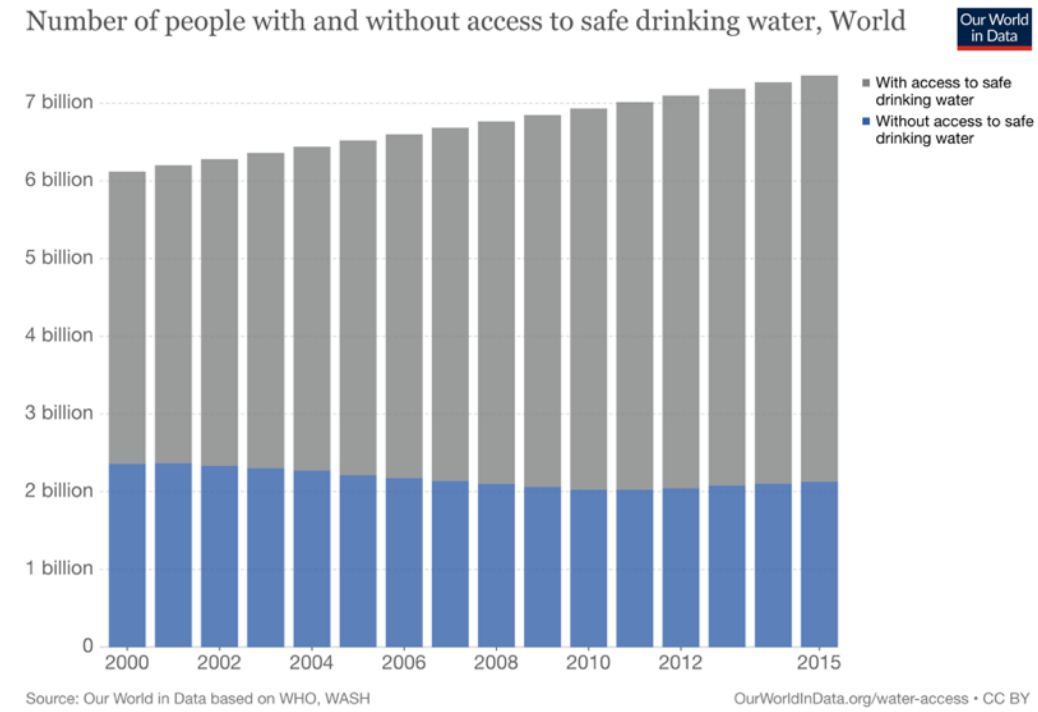


Figure 1.3 Number of people with and without access to safe drinking water.

(Source: Ritchie & Roser, 2018)

Even though there is an increasing amount of historical data in WWTPs, most information in the data will remain unexploited. According to the operator's point of view, the main reason for this is the high dimensionality of the data, where traditional analysis tools cannot be used (Durrenmatt, 2011). As a result, a large amount of data is lost. Various methods and tools such as big data analytics, statistical analysis, deep learning, and artificial intelligence (AI) applications extract information hidden in the data. Furthermore, this would assist the operator in further optimization of the WWTP, improve the effluent quality, reduce the human errors in operating processes, foster the operator's knowledge of the plant processes, and provide other supporting information. Finally, the main goal of a WWTP is to ensure the efficiency of wastewater treatment operation, which is tremendously essential for community health and the environment. Advanced tools and techniques to improve wastewater treatment operations were introduced in this study for analyzing, modeling, optimizing, and forecasting wastewater treatment quality.

1.2 Problem Statement

A literature review on big data management in wastewater treatment operation indicates four principle wastewater management problems that need to be solved in this study: (1) lack of research on big data management from WWTP, (2) lack of comprehensive statistical analysis of the data, and (3) lack of real-time predictive models to forecast effluent quality.

1.2.1 Lack of Research on Big Data Management from WWTPs

After researching publications in big data management in wastewater treatment facilities from various resources such as Google Scholar®, Scopus®, and Web of Science®, there have been limited studies (Ghernaout et al., 2018; Romero et al., 2017). A large amount of data is generated from various wastewater treatment operations every day. Big data should be exploited to enhance the operating systems. Unfortunately, due to the lack of specialized tools, operators and engineers

cannot extract meaningful and valuable information from the massive amount of high-dimensional data.

1.2.2 Lack of Comprehensive Statistical Analysis of Wastewater Treatment Data

Several studies have analyzed energy flow and influent in wastewater treatment facilities to monitor, assess, and model the WWTPs (Martin & Vanrolleghem, 2014). In addition, many publications have illustrated the usefulness of statistical analysis models for WWTP optimization (Cheng et al., 2019; Newhart et al., 2019); operation (Garbowski et al., 2018); analysis (Pantsar-Kallio et al., 1999; Taheriyoun & Moradinejad, 2015; Zhang et al., 2019) and control (Harrou et al., 2018; Maiza et al., 2013). However, few studies have been performed on finding patterns, determining the relationship of each parameter, and selecting meaningful information from big data in wastewater treatment operations with the use of advanced analytics. As a result, big data is underutilized.

1.2.3 Lack of Predictive Models to Forecast Effluent Quality

The water quality predictions in WWTPs were attempted using advanced machine learning tools and techniques. Nonetheless, without the pretreatment of big data, the forecast may not be accurate. Also, previous studies focused on traditional deterministic modeling methodology (Boyd et al., 2019; Huang et al., 2016; Khademikia et al., 2016; Pisa et al., 2019). The novel deep learning method, an RNN, is rarely applied. The conventional approaches might be accurate in predicting if the system is fixed and all the parameters are determined. Furthermore, it is time-consuming, intrusive, and limited by small data analysis.

1.2.4 No Logistics for a Real-Time Model for WWTPs

Real-time data reflects the status of an operational system. The common characteristic of the data is the strict time constraint (Wu et al., 2006). Real-time information is associated with a timestamp and life cycle, and they are only valid for the responding sampling time. Deep learning algorithms with real-time modeling would be a reliable tool for early warning of potential operational upsets and subsequent effluent quality permit violations. There is no study on the development of logistics for real-time modeling using RNNs to help operators for monitoring, detecting fault operation systems in WWTPs.

1.3 Research Objectives

The main objective of this paper is to enhance the operation and performance of WWTPs through big data management and model prediction with AI applications. The study has the following four specific objectives:

(1) To develop analytics for big data from a WWTP;

(2) To interpret analytics results for extracting meaningful information;

(3) To develop deep learning models for forecasting effluent quality from historical wastewater treatment data, which include train and evaluate the models to acquire the most effective algorithms; and

(4) To establish logistics for a real-time self-learning AI system for monitoring and detecting problems during WWTP operation.

1.4 Research Scope and Methodology

Four principal tasks in this study are discussed below.

Task 1: Big data analytics frameworks in wastewater treatment operation.

This task involves collecting data, understanding the processes in wastewater treatment, visualizing data, selecting meaningful information, and developing big data analytics procedures.

Task 2: Statistical analysis techniques to obtain a pattern and meaningful information.

This task applies various statistical methods such as descriptive statistical analysis, correlation coefficient, box plot, normal distribution, and hypothesis testing to find a relationship between parameters, a pattern of data, and insight of information.

Task 3: DNNs model development for prediction.

This task is to develop a predictive model by implementing advanced modeling techniques, Recurrent Neural Networks (RNNs). The methods include preparing the data, selecting the train and test dataset, build a simple RNNs model and the Long Short-Term Memory (LSTM) model with a different number of hidden layers, train the models to predict an output result, and compare and evaluate the models. The parameters to be predicted will be Total Phosphorus (TP), Total Suspended Solids (TSS), and NH_3/Total Nitrogen (TN) in addition to BOD_5 (Biochemical Oxygen Demand) and lastly, Sludge Volume Index (SVI). Control of dissolved oxygen (DO), sludge and energy management logistics may be attempted from the big data.

Task 4: Propose logistics for a real-time prediction model to detect fault errors in wastewater treatment operations.

Unless a WWTP allows access to real-time big data and incorporates the AI model into their Supervisory Control and Data Acquisition (SCADA) system, it is impossible to implement a real-time prediction model to the existing system. Therefore, this task proposes logistics for

real-time model development in WWTPs to help a WWTP's operator for failure detections beforehand.

1.5 Organization of the Dissertation Proposal

The thesis proposal is organized as follows:

Chapter 2 presents a comprehensive review of previous studies on big data in the wastewater treatment sector, statistical analysis in wastewater treatment systems, and the development of prediction models in WWTPs.

Chapter 3 presents big data management with statistical analysis. Big data analytics processes include data collection, data understanding, data preparation, data mining, evaluation, and deployment. Data was collected from the Nine Springs WWTP in Madison Metropolitan Sewerage District (MMSD), Wisconsin. This data is comprehensively studied through data preprocessing techniques contain data cleaning, data integration, data transformation, and data reduction. Finally, statistical analysis techniques extracted meaningful patterns and information to obtain the appropriate dataset.

Chapter 4 presents the development of model predictions using RNNs. After the big data is extracted to a manageable size and the statistical preprocessing technique is implemented to select meaningful information, the prediction efficiency of wastewater effluent quality will improve significantly. The results of traditional RNN models and RNN-LSTM models were compared and evaluated to choose an optimization algorithm for implementation.

Chapter 5 proposes the subsequent work plan to develop an RNN model using different parameters, develop logistics for monitoring, detecting, and proactive maintenance, assist better decision making, and optimize wastewater treatment facilities.

Chapter 6 presents the prediction model of the Sludge Volume Index (SVI) along with data analysis and an Explainable AI algorithm to interpret the result.

Figure 1.4 presents the organization of this study.

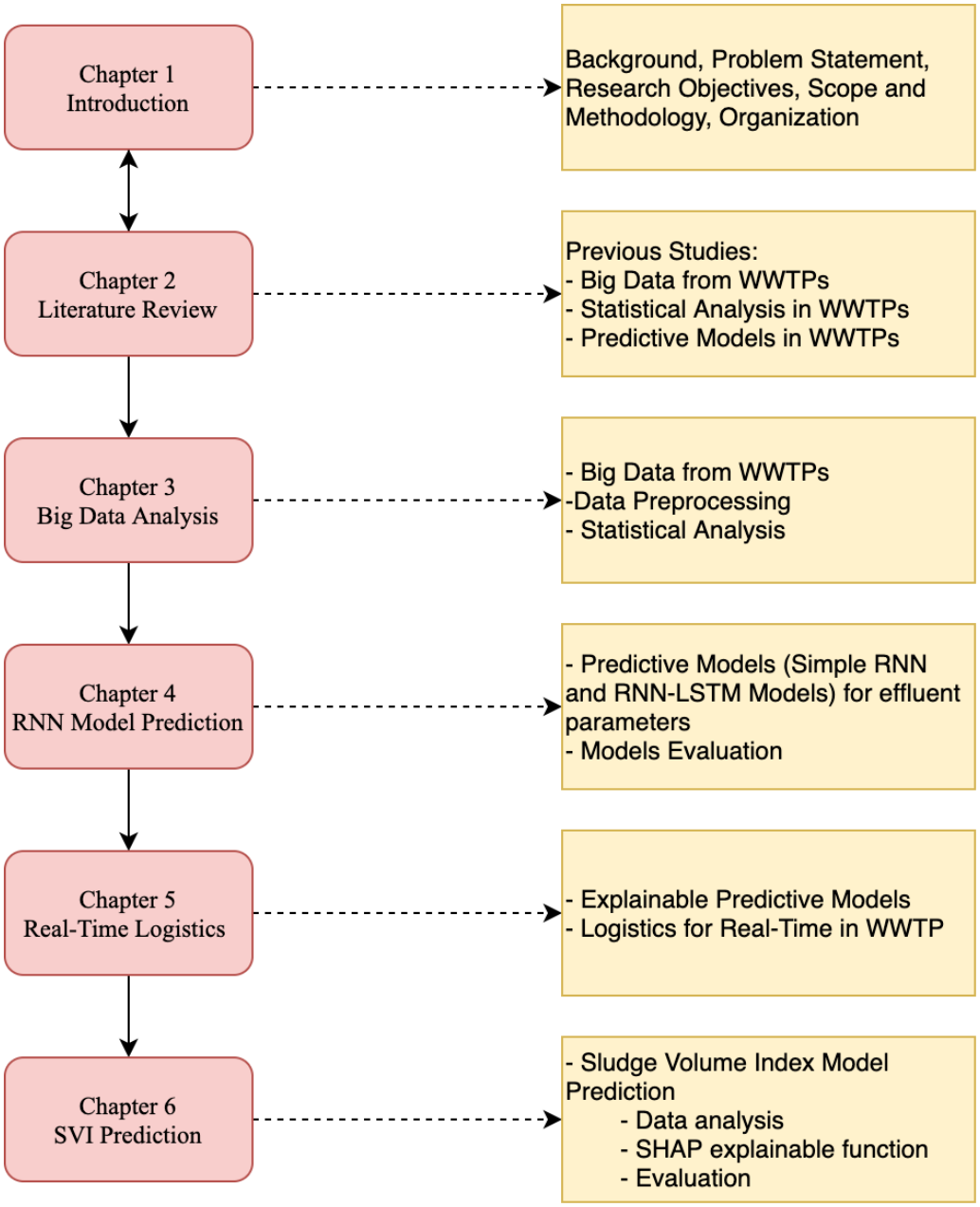


Figure 1.4 Organization of the preliminary dissertation proposal.

2. LITERATURE REVIEW

2.1 Previous Studies of Big Data from WWTPs

2.1.1 Big Data Basics

Big data is the information assets that can be characterized by 'Volume,' 'Velocity,' 'Variety.' Moreover, specific analytical methods are required to transform the data into 'Value' (Mauro et al., 2015). Su (2018) proposed the three V principles of big data – Volume, Velocity, and Variety – as follows (Figure 2.1):

- Volume: Large amounts of datasets with sizes of terabytes to zettabytes.
- Velocity: Large amounts of data at the high captured rate and the rate of data flow. The high speed and time to act based on these data streams will often be very short.
- Variety: Data comes from different data sources and various formats obtained from the web, texts, sensors, etc. Structured data includes a database table, semi-structured data such as XML data, and unstructured data is contained text, images, video streams, audio statements, and more. There is a shift from traditional data analytics of sole structured data to new technical methods analyzing unstructured data or combining the two.

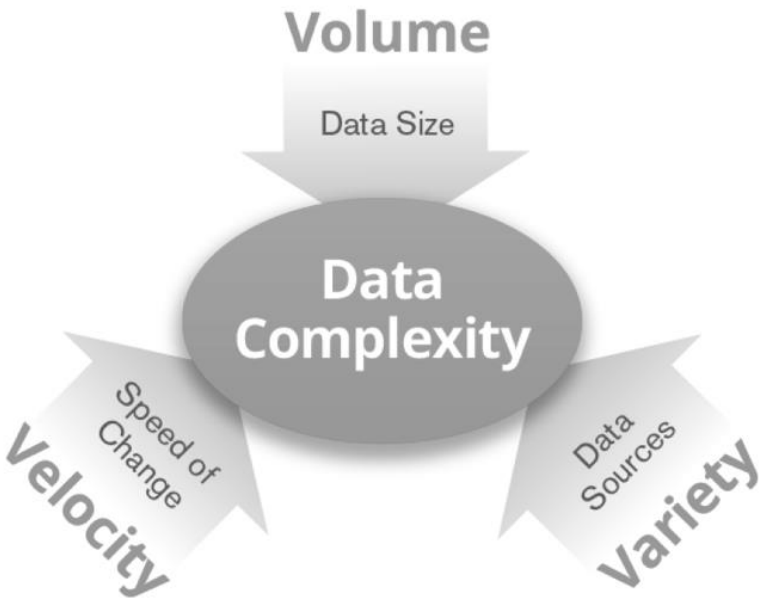


Figure 2.1 The three V principles of big data. (Source: Su, 2018)

Mauro et al. (2015) have studied the definitions of big data associated with four themes, Information, Technology, Methods, and Impact, from various resources (Table 2.1).

2.1.2 Type of Big Data Analytics

Riahi & Riahi (2018) studied concepts, types, and technologies of big data and big data analytics. Big data means data that exceeds the usual storage, processing, and computing capacity of traditional databases and data analysis techniques. Therefore, big data requires techniques that can be employed to analyze meaningful patterns from large-scale data.

Big data analytics is collecting, organizing, analyzing large datasets to discover different patterns and other useful information. It is a set of techniques that require new types of integration to reveal insights from extensive data that are different from the normal ones, more difficult, more complex, and enormous. The main goal is to solve problems in better and effective ways.

There are the following four main types of data analytics:

Table 2.1 The definitions of big data related to four themes. (Source: Riahi & Riahi, 2018)

Definition	I	T	M	P
High volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.	x		x	x
The four characteristics defining big data are Volume, Velocity, Variety and Value.	x			x
Complex, unstructured, or large amounts of data.	x			
Can be defined using three data characteristics: Cardinality, Continuity and Complexity.	x			
Big data is a combination of Volume, Variety, Velocity and Veracity that creates an opportunity for organizations to gain competitive advantage in today's digitized marketplace.	x			x
Extensive datasets, primarily in the characteristics of volume, velocity and/or variety, that require a scalable architecture for efficient storage, manipulation, and analysis.	x	x		
The storage and analysis of large and or complex data sets using a series of techniques including, but not limited to: NoSQL, MapReduce and machine learning.	x	x	x	
The process of applying serious computing power, the latest in machine learning and artificial intelligence, to seriously massive and often highly complex sets of information.	x	x	x	
Data that exceeds the processing capacity of conventional database systems.	x	x		
Data that cannot be handled and processed in a straightforward manner.	x		x	
A dataset that is too big to fit on a screen.	x			
Datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze.	x	x	x	
The data sets and analytical techniques in applications that are so large and complex that they require advanced and unique data storage, management, analysis, and visualization technologies.	x	x	x	
A cultural, technological, and scholarly phenomenon that rests on the interplay of Technology, Analysis and Mythology.		x	x	x
Phenomenon that brings three key shifts in the way we analyze information that transform how we understand and organize society: 1. More data, 2. Messier (incomplete) data, 3. Correlation overtakes causality.	x		x	x

(1) Descriptive Analytics (It consists of asking the question: What is happening?)

It is a primary stage of data processing that generates a set of historical data. Data analytic methods classify data and help discover patterns that provide insight. Descriptive analytics offers future probabilities and trends and presents an idea about what might happen in the future.

(2) Diagnostic Analytics (It consists of asking the question: Why did it happen?)

Diagnostic analytics considers the root of a problem. It is applied to determine why something happened. This type tries to find and understand the reasons for happenings and behaviors.

(3) Predictive Analytics (It consists of asking the question: What is likely to happen?)

It implements historical data to predict the future. It is all about prediction or forecast. Predictive analytics applies many tools like data mining, deep learning, and artificial intelligence to analyze data and develop the model of what might happen.

(4) Prescriptive Analytics (It consists of asking the question: What should be done?)

It is dedicated to finding the right action that should be taken. Descriptive analytics contains historical data, and predictive analytics helps forecast what might happen. Prescriptive analytics employs these parameters to find an optimized solution.

2.1.3 Data from WWTPs

Big data usually refers to collecting information that generates too large to process, analyze, and manage with the traditional software on time. The main objective of data management is to find a new technology or technique for collecting, storing, analyzing, managing, and mining a massive dataset. Big data analytics is a new concept and is not widely exploited, especially in wastewater treatment operations. The wastewater treatment process is typically complicated, in which the

applications of big data will have more outstanding features than in other fields such as finance, electricity, and biology (Ning, 2018).

WWTPs have many water quality sensors that generated a vast amount of data every day. There is an insight in harnessing the historical and real-time data to complement traditional operational decision support systems (Ghernaout et al., 2018). Developments in data analysis allow for the large amount of data generated to be transformed into informative insights and decision support in a fraction of the time it would take a human.

Ghernaout et al. (2018) provided five keys to analyze big data in wastewater treatment systems and avoid the traps of bad data.

(1) Data quality rather than quantity

Even the most advanced analytics may have errors in data measurements, whether it is noisiness, drift, or interpolations. Bad data will make a model worthless resulting from errors in measurement devices and analyzers. Confidence in sensors and analyzers may be reached by following three fundamental steps. Table 2.2 gives an abstract of these primary stages.

Table 2.2 Three stages for gaining confidence in sensors and analyzers. (Source: Shaw, 2017)

	Three stages for better manipulating sensors and analyzers
Stage #1: Cleaning them	Wastewater treatment is an especially fouling environment and not the best place to put scientific equipment. Operators frequently underestimate how quickly sensors become fouled. Go for auto-cleaning whenever possible and avoid installing anything in raw sewage or primary effluent unless you really need the measurement because both areas are particularly prone to fouling. Mixed liquor is an easier place to take measurements, and final effluent is the easiest place of all. Water treatment systems usually are less fouling, but sensors still need periodic cleaning.
Stage #2: Calibrating them	This is generally understood, although the frequency of calibration, particularly for sensors that tend to drift, typically is shorter than ideal.
Stage #3: Validating them	This may be the action overlooked by most instrumentation suppliers. Analytics to validate the measurements, particularly during calibration, frequently need more attention.

(2) Measuring useful items

WWTPs need fundamental measurements such as dissolved oxygen (DO) in the aeration basins, airflow to each aeration zone, and electricity use by blowers. Still, operators need to be

careful not to take measurements that are not principally useful. The plants may spend a large amount of money measuring ammonia and nitrate all over a treatment plant. However, it is not employed. These measurements will possibly be disregarded, and the systems neglected.

(3) Dynamics rather than steady-state

WWTPs get significant daily variations in flows and concentrations; as a result, they are considered dynamic systems. It is required to quantify and analyze the dynamics of the operations to comprehend the treatment systems (Shaw, 2017).

(4) Different timescales

Dynamics in data are required to consider different timescales such as daily, variations, weekly trends (especially weekend versus weekday differences), and seasonal shifts. Therefore, data analytics conditions are different and need to be carefully studied.

(5) Handling outliers and extraordinary events

In the process of big data analytics, it is necessary to identify and eliminate outliers, whether they are either wrong measurements or unusual, thus something to ignore. However, WWTPs aim to keep the processes stable in response to abnormal events, such as shock loads, toxins, or seasonal change. Therefore, these have to be carefully considered outliers and decide what to do, rather than throw them away.

2.1.4 Big Data for Better WWTP Management

All the existing technologies aim to expand water resources and deliver them to end-users (Ghernaout et al., 2018). Big data analytics can help optimize the balance between performance and reliability to prevent human-made disasters, such as sudden drops in water quality, which might not be detected until effects are realized. In addition, big data can help WWTPs understand trends and patterns that will affect designing and planning an adaptive and responsive wastewater

system. Big data predictive analysis can also help operators act effectively to deal with an upcoming event.

2.2 Previous Studies on Statistical Analysis in Wastewater Treatment Systems

Cortés-Martínez et al. (2016) applied statistical analysis of the concentration of pollutants in the influent. The parameters, biochemical oxygen demand (BOD) and chemical oxygen demand (COD), were analyzed. Descriptive statistics were applied to the data to analyze pollutants in the influent in the treatment plant to establish control of contaminants before being discharged into the municipal sewage. Table 2.3 shows the classification of concentration in wastewater (Metcalf & Eddy et al., 2003).

Table 2.3 The classification of concentration in domestic wastewater.

Parameters	**Units**	**Concentration**		
		Weak	Medium	Strong
Total Suspended Solids (TSS)	mg/L	130	195	389
Biochemical Oxygen Demand (BOD_5)	mg/L	133	200	400
Nitrogen (total as N)	mg/L	23	35	69
Free ammonia	mg/L	14	20	41
Phosphorus (total as P)	mg/L	3.7	5.6	11.0

Cortés-Martínez (2016) applied descriptive statistical analysis to analyze a vast database of the characterizations in the influent wastewater treatment system. The results provide a basis for future comparisons to identify significant deviations in the concentration of pollutants. Thus, it is possible to detect potential offenders.

Multivariate statistical techniques have been developed for improving, assessing, and monitoring wastewater treatment facilities. The methods include factor analysis, principal component analysis (PCA), cluster analysis, and multiple regression analysis (Bakia et al., 2019). Other approaches applied latent variable (LV) techniques, such as PCA and projection latent structures (PLS), to monitor and forecast the influent and effluent parameters in wastewater treatment facilities.

Summaries of the papers applying statistical analysis are summarized below:

- Garbowski et al. (2018) calculated the value of the treatment plant reliability factor (RF), the quotient of the average concentration of an individual pollutant indicator in the treated sewage, and the permissible value in the wastewater discharged to the natural resources. There are five essential pollutants applied to evaluate the contaminants removal efficiency: biochemical oxygen demand (BOD_5), chemical oxygen demand (COD), total suspended solids (TSS), total nitrogen (TN), and total phosphorus (TP).
- Pantsar-Kallio et al. (1999) proposed multivariate data analysis of critical pollutants in sewage samples using principal component analysis (PCA) and partial least squares (PLS) methods. With multivariate techniques, effects of the lifestyle of residents, day of the week, and sampling time or weather on the pollutant levels were determined.
- Cheng et al. (2019) developed nonlinear data-based techniques to identify potential anomalies that would trigger adverse events in WWTPs. Besides, influent conditions (ICs) as initial states of inflow fed to WWTPs were monitored. The goal is to operate WWTPs with optimized efficiency. The nonlinear, non-Gaussian, non-stationary, auto-correlated, cross-correlated, hetero-skedastic, case-specific nature of multivariate environmental datasets was implemented for data-driven decisions. Furthermore, a seven-year

multivariate influent conditions time series was introduced with exploratory analysis performed to reveal temporal behaviors and statistical properties.

Harrou et al. (2018) reviewed publications about statistical analysis techniques to improve the efficiency of WWTPs as shown below:

- Wang et al. (2017) proposed a statistical approach based on combined principal components analysis (PCA) and multiple regression to model a WWTP.
- Amaral and Ferreira (2005) applied a partial least square (PLS) regression for the activated sludge process (ASP) monitoring.
- PLS methods have been applied to predict sludge sedimentation properties by monitoring the parameters affecting effluent quality and filamentous bulking in ASPs (Mujunen et al., 1998).
- PCA and its extensions have been widely used for statistical modeling and monitoring WWTPs (Liu et al., 2014; Huang et al., 2012; Villez et al., 2008; Lee et al., 2004; Rosen and Lennox, 2001; Lee and Vanrolleghem,2003).

This study aims to enhance the operation and performance of WWTPs through the development of a deep belief network (DBN) model and a one-class support vector machine (OCSVM) to model the complex variables of data from WWTPs and separate normal from abnormal features. The authors compared the results from the DBN-OCSVM approach to K-NN-EWMA and K-Means algorithms. The limitation is that when data is very noisy, the quality of the developed detection algorithm was affected to a greater extent. Thus, it is essential to develop a robust statistical approach that can handle noisy data to detect faults during the operation of WWTPs better.

2.3 Previous Studies of the Development of Predictive Models in WWTPs

Several forecasting models have been so far developed to monitor wastewater treatment processes. However, traditional predictive models have complex structures and are involved in various numbers of parameters that must be identified (Harrou et al., 2018). The consensus is that the models are complex and unsuitable for plant operation. For example, the model ASM1 comprises 13 nonlinear differential equations requiring 19 parameters that are hard to estimate (Dochain & Vanrolleghem, 2000) and specific to wastewater characteristics. The difficulty of the previous modeling technique also includes a heavy computational burden for the simulation and design process (Vanrolleghem et al., 1999). Due to the challenges in the simulation of such complex systems, numerous data, and heavy computation, many modeling techniques have been developed, such as the autoregressive (AR) model, artificial neural network (ANN), genetic algorithm, multivariate analysis, and so on.

2.3.1 Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a method to forecast future behavior from the previous historical data. It can be written as "ARIMA (p, d, q)," where parameter p is the order of the autoregressive (AR) model, parameter d is the degree of differencing, and parameter q is the order of the moving average (MA) model (Boyd et al., 2019). Box and Jenkins (1970) created the iterative steps of the ARIMA approach include: identification, estimation, and validation, as shown in Figure 2.2. The value (d) is observed in the identification process. The assessment step includes selecting parameters and coefficients from the training set. The next step is to validate by processing the time series using the optimal (p, d, q) values. Many researchers later applied the Box-Jenkins model to predict time series, so it is the fourth step in the Box-Jenkins approach.

Boyd et al. (2019) proposed the ARIMA method to forecast the influent of WWTPs. The dataset was split into training and validation sets. Three parameters, p, d, and q, were manually configured for the training set. The optimal combination of *(*p, d, q*)* needed to be calculated to help calibrate the model for better performance. However, Boyd et al. mentioned that the combinations in a large dataset might lead to overfitting in the model because the higher p and q are, the higher the number of parameters is. As a result, the model is prone to overfitting. Other methods should be used to penalized overfitting when the values of p and q are high. Therefore, Boyd et al. recommended that nonlinear models, such as neural networks, would instead perform better than ARIMA for forecasting purposes.

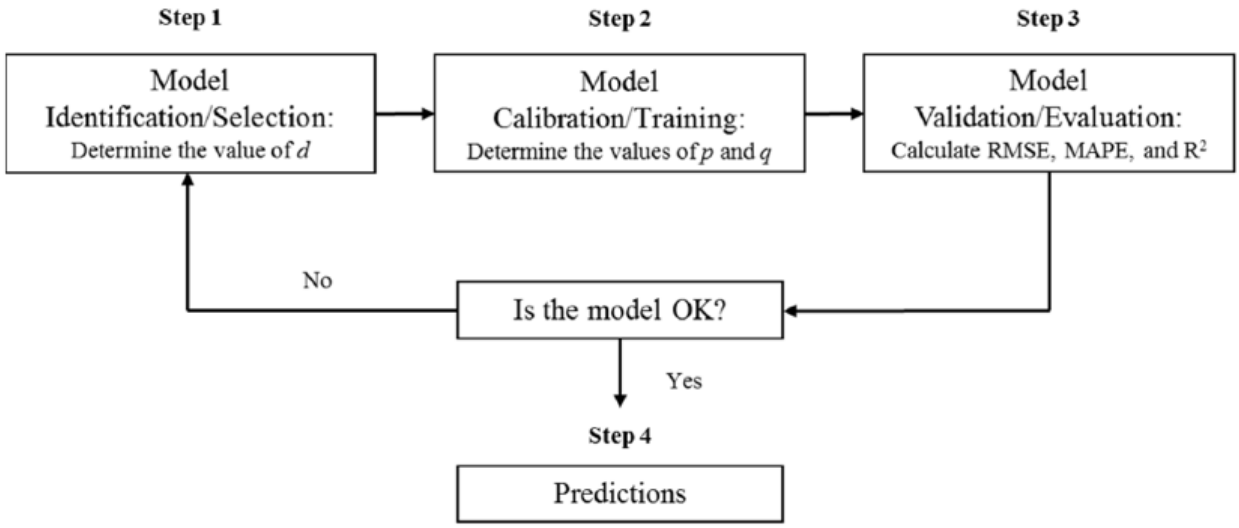


Figure 2.2 The iterative steps of the ARIMA approach. (Source: Box and Jenkins, 1970)

Zhao et al. (2020) reviewed four aspects of the application of AI to wastewater treatment. The elements include technology, economics, management, and wastewater reuse by bibliometric analysis and systematic review.

Figure 2.3 shows that the ANN and FL models are the most widely used methods in single models, and the NF and ANN-GA models are much more frequently used in hybrid models. The number

of applications of the single ANN model increased 93% more in 2015–2019 than in 2010-2014. The number of uses of the hybrid ANN-GA model increased four times more in 2015–2019 than in 2010–2014. The classification of the AI technologies involved in wastewater treatment is shown in Figure 2.4.

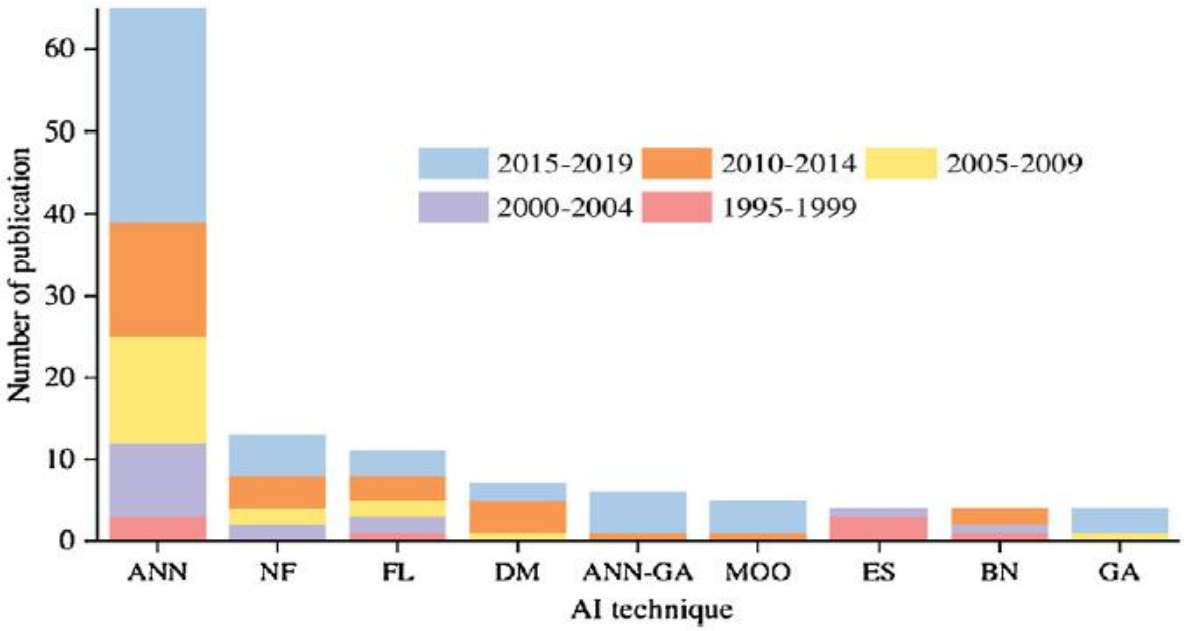


Figure 2.3 Frequency and trend of AI techniques applied to wastewater treatment during 1995–2019. (Source: Zhao et al., 2020)

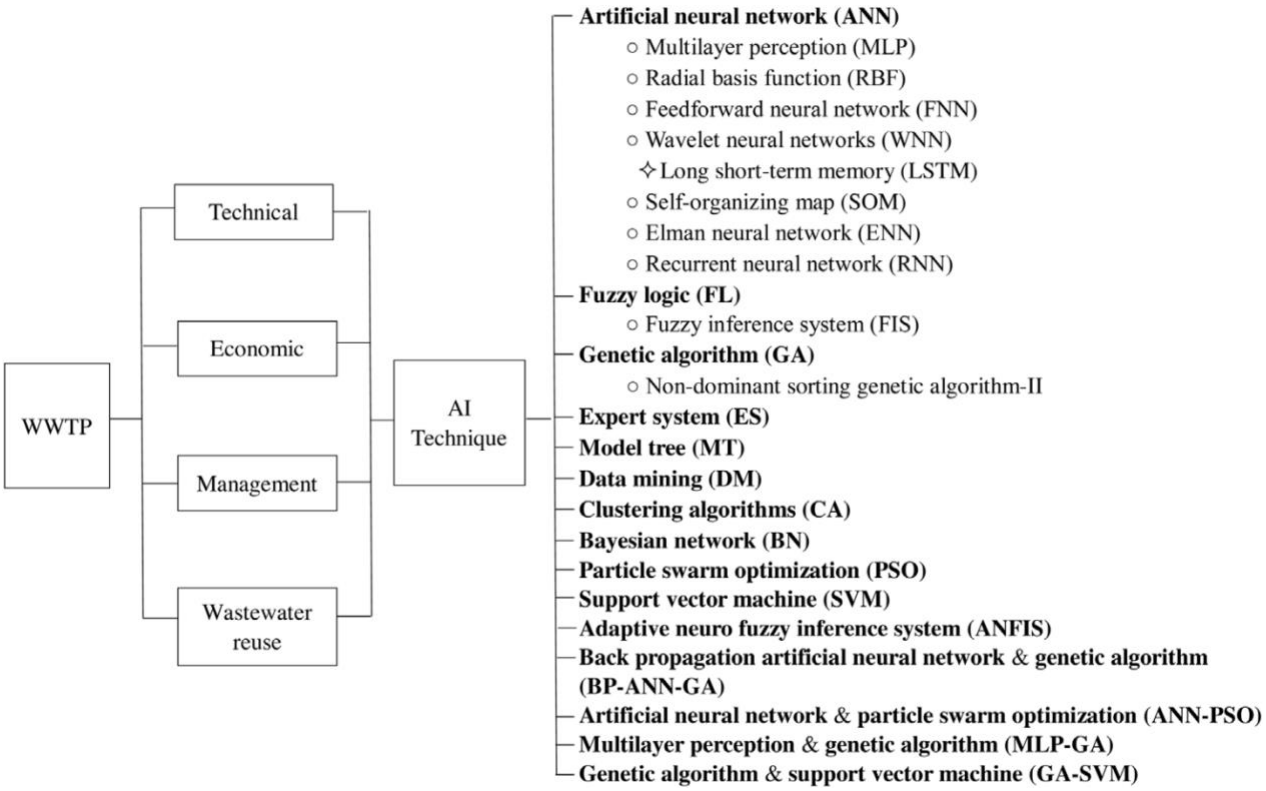


Figure 2.4 Classification tree of AI technology used in wastewater treatment.

(Source: Zhao et al., 2020)

ANN, ANN-GA, DM, MT, SVM, NF, and ANFIS are major AI models used in the management of wastewater treatment to help monitor, predict, evaluate, and analyze wastewater treatment operations. AI technologies such as ANN, ANFIS, NF, RL, and MOOC are applied to improve wastewater treatment processes' efficiency and reduce costs by controlling the flow rate, influent, effluent, monitoring systems, and automation. ANNs, FLs, DMs, and GAs combining with NF and ANN-GA were the most generally implemented single models for WWTPs. They could provide higher precision and lower inaccuracy.

In summary, the following four essentials are worth mentioning for future research:

(1) Better AI models are required to generate optimal operation systems, higher contaminants removal, and lower operating cost, especially under variation circumstances.

(2) The prediction capability of AI technologies should be reinforced by including significant parameters of wastewater treatment processes to ensure effluent quality standards.

(3) Future research should use a larger amount of historical data or online data to support AI models to perform faster and more precisely in real applications of wastewater treatment.

(4) A model should combine with significant aspects, including technology, economics, management, and wastewater reuse. Thus, a model would help address contaminant removal, cost reduction, water reuse, and management tasks simultaneously.

2.3.2 Artificial Neural Network (ANN)

The neural system of humans stimulates the fundamental concept of ANN. Each neuron is connected to develop a layered structure from inputs to outputs through a few hidden layers (Haider et al., 2019). Each layer represents several hidden neurons that are characterized by the activation function, which are applied to obtain the neuron output from the input. The layers are connected through weights and biases, called the network's coefficients.

Haider et al. (2019) briefly introduced the basic concepts of ANN, Recurrent Neural Networks (RNNs), and Long Short-Term Memory Neural Networks (LSTM). Figure 2.5 shows the basic structure of an ANN. It includes a set of inputs ($x_1, x_2, ..., x_R$), an output (y_o), and two hidden layers, δ_i and δ_i'. The hidden layers are called threshold or quantization functions.

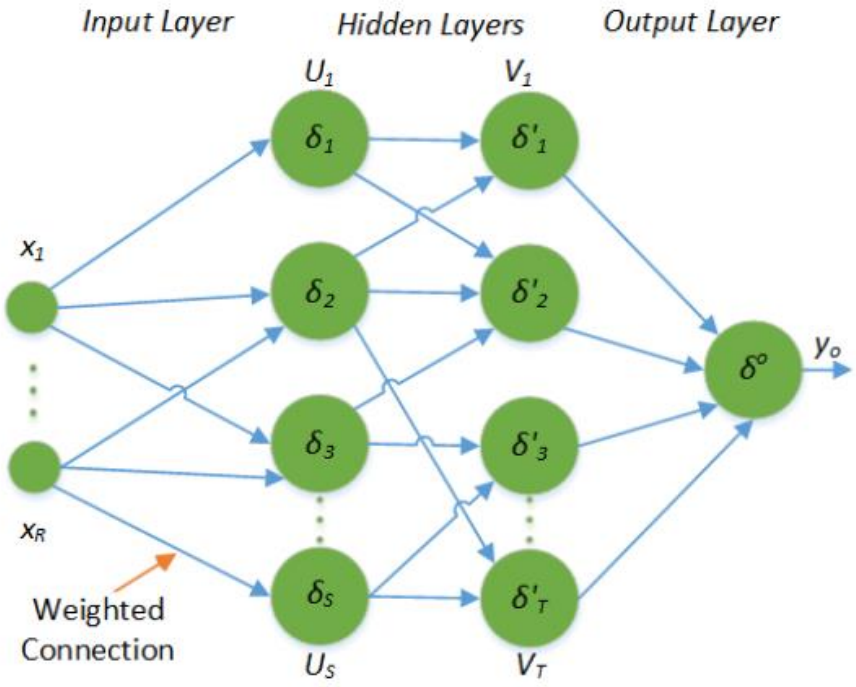


Figure 2.5 Basic structure of an Artificial Neural Network (ANN). (Source: Haider et al., 2019)

ANNs can model nonlinear systems, and thus, many researchers developed predictive modeling systems. The model is first trained by data of a learning algorithm, in which the network's coefficients are updated iteratively. The trained model is later used for attaining the desired patterns. One of the most general learning algorithms is backpropagation; however, the error is propagated backward. Thus, many ANN architectures are developed, such as feed-forward, recurrent neural networks, and Long Short-Term Memory.

Feed-forward neural networks have been successfully implemented to solve problems requiring the computation of a static function, such as a function whose output depends only upon the current input and not previous inputs (Qiao et al., 2011). However, in the wastewater treatment system, there are many variations in conditions that change data. Therefore, feed-forward neural networks are not suitable for the wastewater treatment system.

An RNN is an extension of feed-forward neural networks with the capability to handle variable-length sequence inputs. They contrast with conventional feed-forward neural networks, which cannot handle sequential inputs and outputs. In other words, the inputs of the feed-forward model must be independent of each other.

RNNs provide gates to store the preceding inputs and weigh the following information of the previous inputs. This distinct RNN's memory is called recurrent hidden states and provides the RNNs the ability to predict a coming input of new data. The general architecture of RNN is shown in Figure 2.6, where a feedback path exists between the second and first hidden layer.

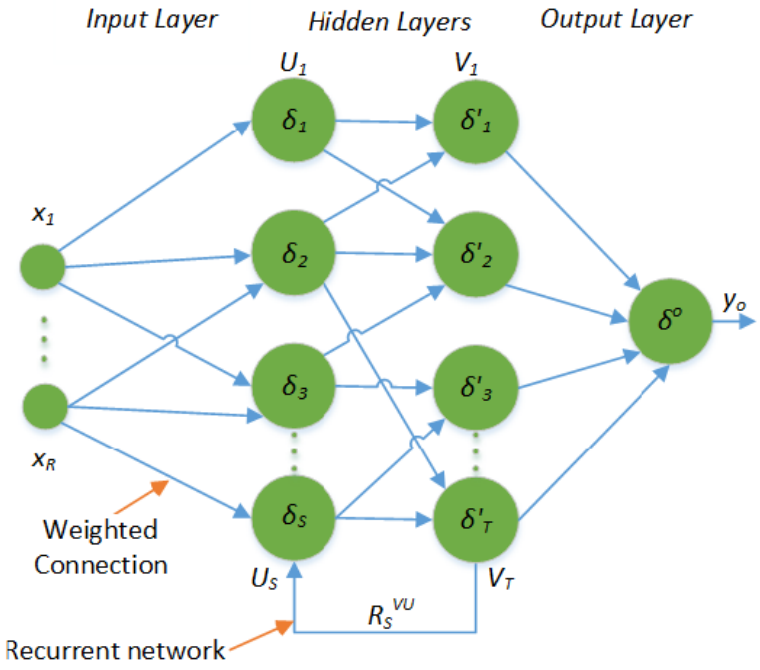


Figure 2.6 Basic structure of a Recurrent Neural Network (RNN). (Source: Haider et al., 2019)

RNNs are altered from the feed-forward system because they not only operate on an input space but also an internal state space. It can be called an iterated function system or a dynamic system. In the general RNN, the neural network outputs are feedback to the same neuron in the preceding layers, signal flow in forwarding and backward directions (Qiao et al., 2011).

One particular kind of RNN is Long Short-Term Memory (LSTM), which can learn long patterns in sequential data. The basic concept of LSTM is shown in Figure 2.7. This particular memory can remember information over a more extended period, enabling reading, writing, and deleting data from their memories.

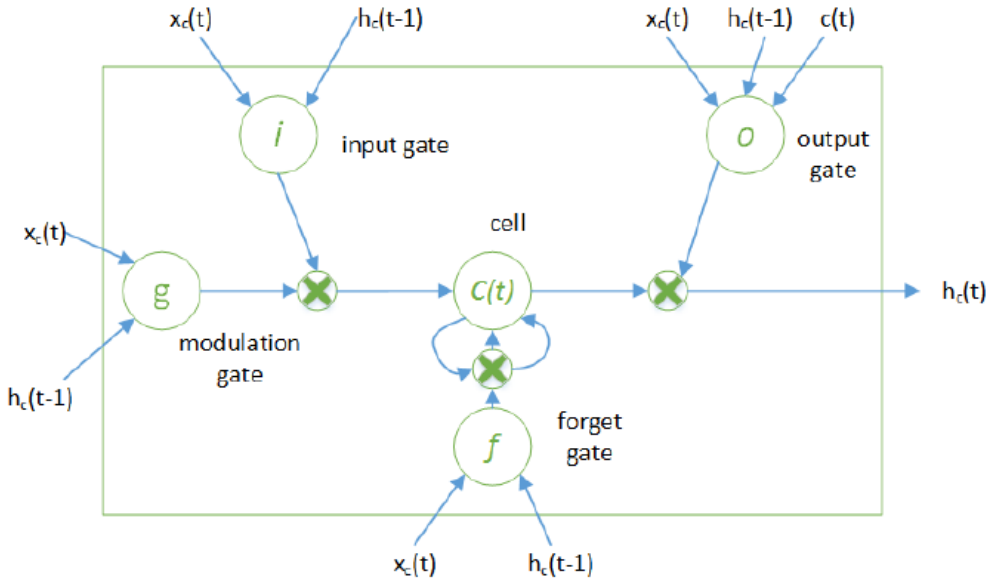


Figure 2.7 Basic structure of a Long Short-Term Memory cell (LSTM).

(Source: Haider et al., 2019)

The LSTM memory is called a "gated" cell, where the word gate means an ability to decide between maintaining or deleting the memory information. The decision to remove or preserve the information is attained based on the weight values assigned to the training procedure. In general, an LSTM model contains three gates: forget, input, and output gates. The forget gate decides to maintain or remove the information, the input gate identifies the size of the new information, which will be added into the memory. Lastly, the output gate monitors whether the existing value in the cell contributes to the output.

2.4 Conclusions and Recommendations

A critical review of the available publications was performed in three crucial concepts, big data analytics, statistical analysis, and predictive model development in WWTPs. The following conclusions can be drawn:

- In the wastewater treatment industry, big data analysis is a relatively new idea and is not extensively exploited. The power of historical and real-time data with advancements in computerized data analysis can be valuable to optimize traditional wastewater treatment systems. The five keys to effectively analyze big data in WWTPs include the importance of data quality, measurement of core parameters, dynamics of data, a difference of timescales, and management of outliers (Ghernaout et al., 2018).
- Several statistical analysis techniques, such as principal component analysis (PCA), cluster analysis, and multiple regression analysis, have been implemented for improving, monitoring, and evaluating wastewater treatment performance (Bakia et al., 2019). However, there are few studies on selecting datasets, finding data patterns, and identifying the relationship of parameters from historical data in wastewater treatment operations.
- Lastly, traditional predictive models have been proposed to optimize wastewater treatment processes. However, the models are complex and unsuitable for data fluctuation in WWTPs (Harrou et al., 2018). The improvement of artificial neural networks (ANNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) Neural Networks are well-suited for sequence data in wastewater treatment systems because of the exceptional learning capability of the networks.

3. BIG DATA ANALYTICS FROM A WASTEWATER TREATMENT

3.1 Abstract

Wastewater treatment plants (WWTPs) use considerable workforce and resources to meet the regulatory limits without mistakes. WWTPs implement the supervisory control and data acquisition (SCADA) system to monitor the operation. The advancement of information technology allowed for collecting large amounts of data from various sources using sophisticated sensors. Unfortunately, operators cannot use the digital data collected from multiple sensors to identify the operation status (Maiza et al., 2013). Due to the lack of specialized tools and knowledge, operators and engineers cannot effectively extract meaningful and valuable information from a large number of datasets. Accordingly, most data collected is wasted. If big data at a WWTP is analyzed using visual and statistical tools, operators will have a much clear understanding of the operation while saving the operation and maintenance costs and reducing the human resources required.

Various data analytics techniques have been developed in the past few years. The methods are efficient for analyzing a vast dataset; they have processing power with high speed of computing and analyzing big data. While these techniques are promising tools to provide meaningful information for operators and engineers, there is no wholly developed study in applying these techniques to assist wastewater treatment operation.

Data analytics processes can immensely transform a large dataset into informative knowledge, such as a predictive model. The use of predictive tools will allow operators, engineers, plant

managers to detect problems earlier, making them proactive rather than reactive in their response (Ghernaout et al., 2018). Ultimately, big data with statistical analysis in the wastewater industry can be applied to enhance the operational performance of the infrastructure.

3.2 Introduction

3.2.1 Background

Big data plays an essential role in many fields of our daily life. The data generation has been tremendously increasing since 2010. Of the world's data, 90% was created in the past two years (Ritchie & Roser, 2018). Water quality sensors in various operational processes have generated a large amount of data. However, the data is often stored digitally and then underutilized.

Wastewater treatment processes involve many operational systems. The objective of wastewater treatment is to remove contaminants from sewage to discharge into natural resources. Treated water must meet effluent discharge permits to protect public health and the environment (Siegrist, 2017). Lack of access to safe water leads to a risk factor for infectious diseases such as cholera, diarrhea, dysentery, etc. Besides, 1.2 people died prematurely in 2017 due to unsafe water (Ritchie & Roser, 2018).

The biggest challenge in data analytics from wastewater treatment is the dynamical behavior of the data. The data is usually complicated and uncertain because of variations from the environmental conditions, changes in the process variables, and fluctuation in the flow rate and concentration of the influent composition (Harrou et al., 2018). Finding insight from historical and real-time data can allow for the better management of WWTPs and advanced operational decision support systems. Thus, big data from WWTPs will be analyzed using statistical tools to develop better operational methods.

3.2.2 Previous Research

Big data is the information assets that can be characterized by 'Volume,' 'Velocity,' and 'Variety.' Moreover, specific analytical methods are required to transform the data into 'Value' (Mauro et al., 2015). Durrenmatt (2011) listed the following typical problems that practitioners must deal with:

(1) Bigger datasets mean not only the amount of data but also data characteristics. Therefore, more effective algorithms, samples, calculation, and parallel processing are needed.

(2) A high-dimensional dataset, i.e., a dataset with many characteristics, on the one hand, increases the search space for model induction, which leads to a phenomenon referred to as the "curse of dimensionality (Verleysen & François, 2005). The higher the dimensionality is, the more intermediate the data points are. The reduction of dimensionality and the inclusion of prior knowledge to remove irrelevant variables should be implemented.

(3) Non-stationary data makes traditional patterns or models invalid. It is primarily a problem for the systems because the processes and the environmental conditions are frequently subject to change. Possible solutions involve frequent updating model patterns or developing adaptive models.

(4) Missing and noisy data can implement clean data techniques such as selecting, filtering, formatting, outlier detection strategies, and so on.

(5) Overfitting means the production of an analysis that provides too closely to a set of data, results in poor performance when the model is implemented to new data. Cross-validation and other modeling strategies can be applied to avoid overfitting.

(6) Discovered patterns must be made understandable by humans and effectively communicated. A large amount of data and visualization techniques are developed for this purpose.

(7) The presence of prior knowledge in a simple way is not available for current techniques and tools, while the consideration of this knowledge is significant for the project accomplishment.

(8) Integration with other systems, for instance, existing process control systems, is a key; a stand-alone detection system is not preferred in most cases.

Big data analytics can help optimize the balance between performance and reliability to prevent human-made disasters, such as sudden drops in water quality, which might not be detected until a violation of the permit or failure of the plant operation occurs. Big data can help operators understand trends and patterns that will affect operational decisions. Big data predictive analysis can also help operators act effectively to deal with an upcoming event.

3.2.3 Shortcoming of Previous Research

To search for ascertaining publications in academic research and journals, specific keywords, big data, wastewater, sewage, and pollutant, were applied in Google Scholar®, Scopus®, and Web of Science®, finding thousands of results. However, there is a limited set of relevant results recorded. Below are lists of related publications:

- Leveraging Big Data Tools and Technologies: Addressing the Challenges of the Water Quality Sector (Romero et al., 2017);
- Applying Big Data in Water Treatment Industry: A New Era of Advance (Ghernaout et al., 2018);
- Sewage Treatment Process Based on Big Data Management Mode (Ning, 2018).

A review of published literature on big data management shows that few studies have been undertaken on (1) understanding patterns of data generated from WWTP, including finding an insight of the data, (2) determining a relationship between important parameters affecting

wastewater treatment quality, and (3) developing the method of data analytics for better wastewater treatment operation.

3.2.4 Study Objectives

The objectives of the study are to:

- Understand an insight and pattern of data in a wastewater treatment facility;
- Determine the relationships between essential parameters affecting wastewater treatment quality; and
- Develop the method of data analytics for better wastewater treatment operation.

3.3 Materials and Methods

The methodology of big data analytics for wastewater treatment operations is described below and shown in Figure 3.1.

(1) Data Collection

Data was collected from the Nine Springs WWTP operated by the Madison Metropolitan Sewerage District (MMSD), Madison, Wisconsin, U.S.A. According to the MMSD 50-Year Master Plan, Figure 3.2 shows the general layout of the treatment and support facilities at the WWTP. Wastewater treatment involves the preliminary treatment, primary clarification, nitrifying activated sludge treatment incorporating biological phosphorus removal, ultraviolet disinfection, and effluent pumping (McGowan & Wang, 2008).

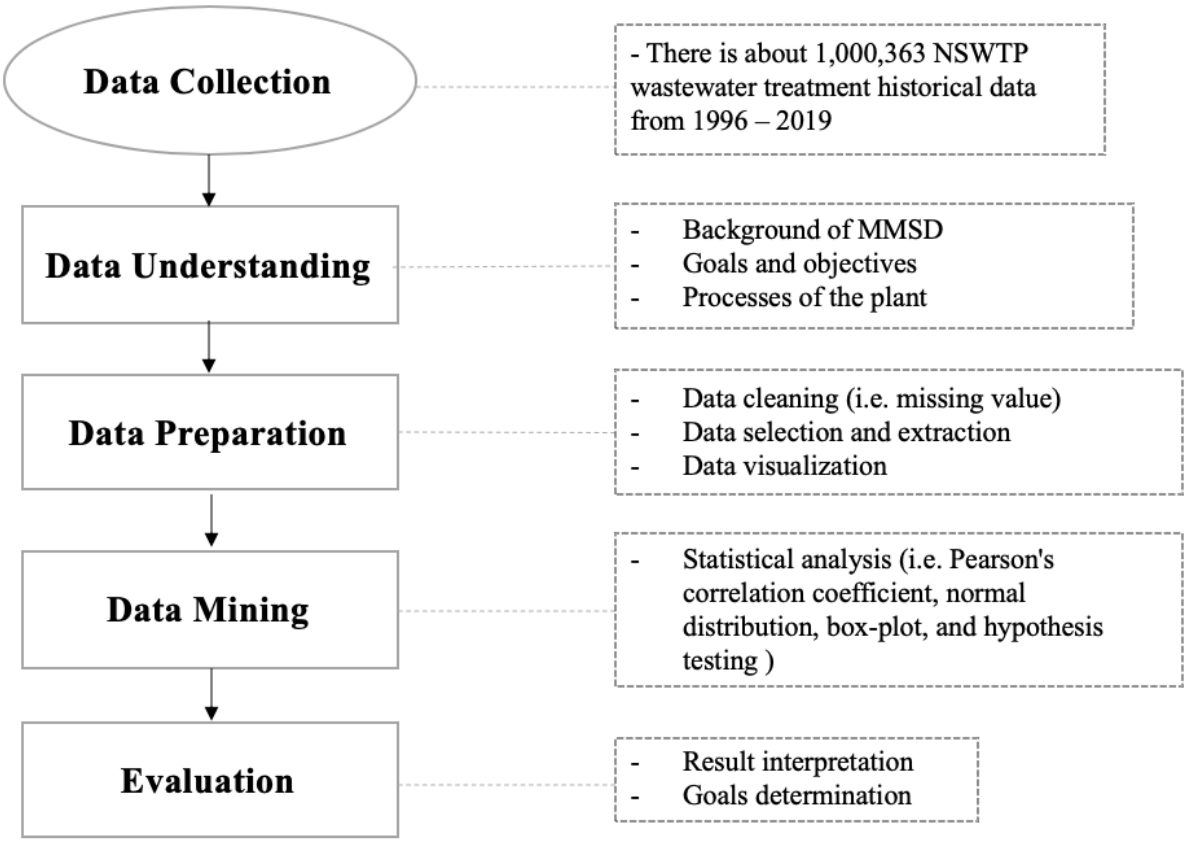


Figure 3.1 The methodology of big data analytics in WWTPs.

Data collected consists of two sets of data. The first data set contains 1,000,363 wastewater treatment historical data (Figure 3.3) from 1996 to 2019. It is a tremendous amount of data because there are many parameters in various processes, and the file size is about 311.6 MB. The data contains many columns, such as 'MeasureCode,' 'LocationCode,' which have different parameters, as shown in Figure 3.4.

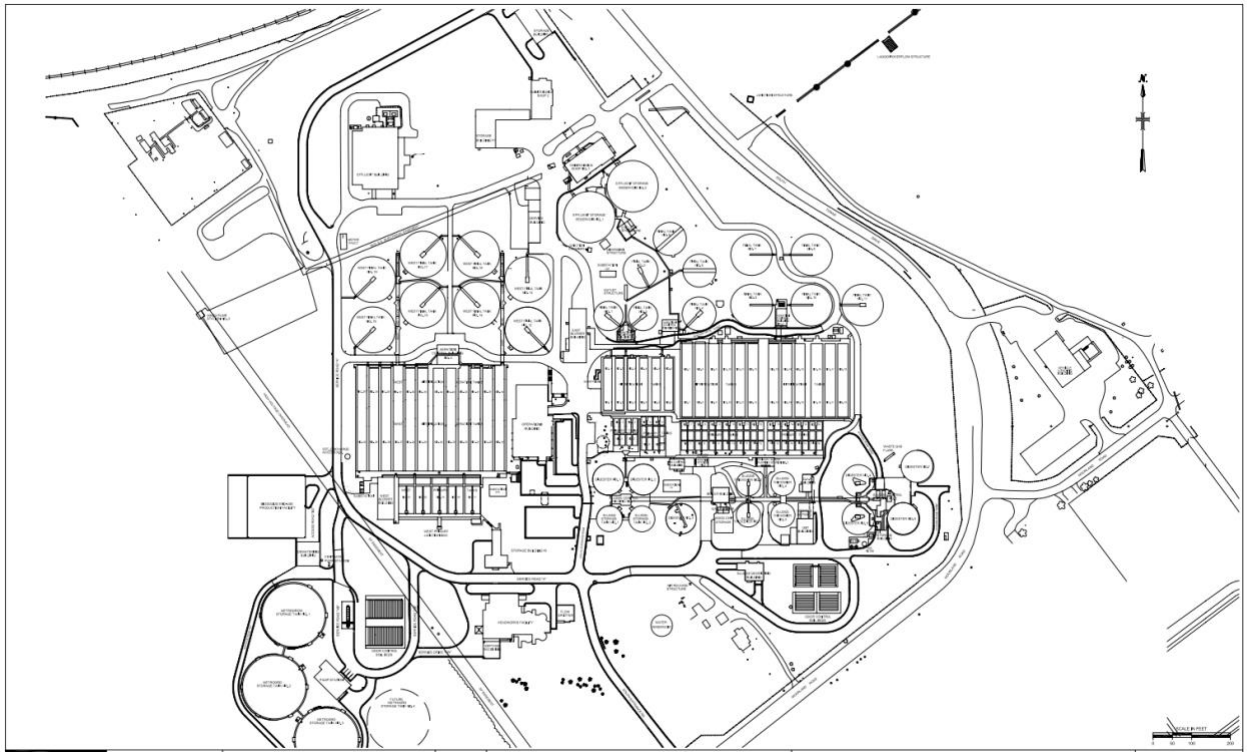


Figure 3.2 Nine Springs Wastewater Treatment Plant. (Source: McGowan & Wang, 2008)

```
In [2]: ResultData = pd.read_csv('ResultAll.csv')

In [3]: print("Number of Data:",len(ResultData))
        Number of Data: 1000363

In [4]: ResultData.info()
        ResultData.head()
        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 1000363 entries, 0 to 1000362
        Data columns (total 13 columns):
        TypeMethodID          991537 non-null float64
        ResultDate            1000363 non-null object
        OriginalResult        994520 non-null float64
        Result                994504 non-null float64
        AnalysisDate          992966 non-null object
        IsBadData             1000363 non-null int64
        SiteCode              1000363 non-null object
        LocationCode          1000363 non-null object
        DescriptionCode       1000363 non-null object
        LabReferenceName      1000284 non-null object
        MeasureCode           1000363 non-null object
        UnitCode              1000363 non-null object
        TypeSampleTypeCode    1000363 non-null object
        dtypes: float64(3), int64(1), object(9)
```

Figure 3.3 The numbers of historical Nine Springs WWTP data from 1996-2019.

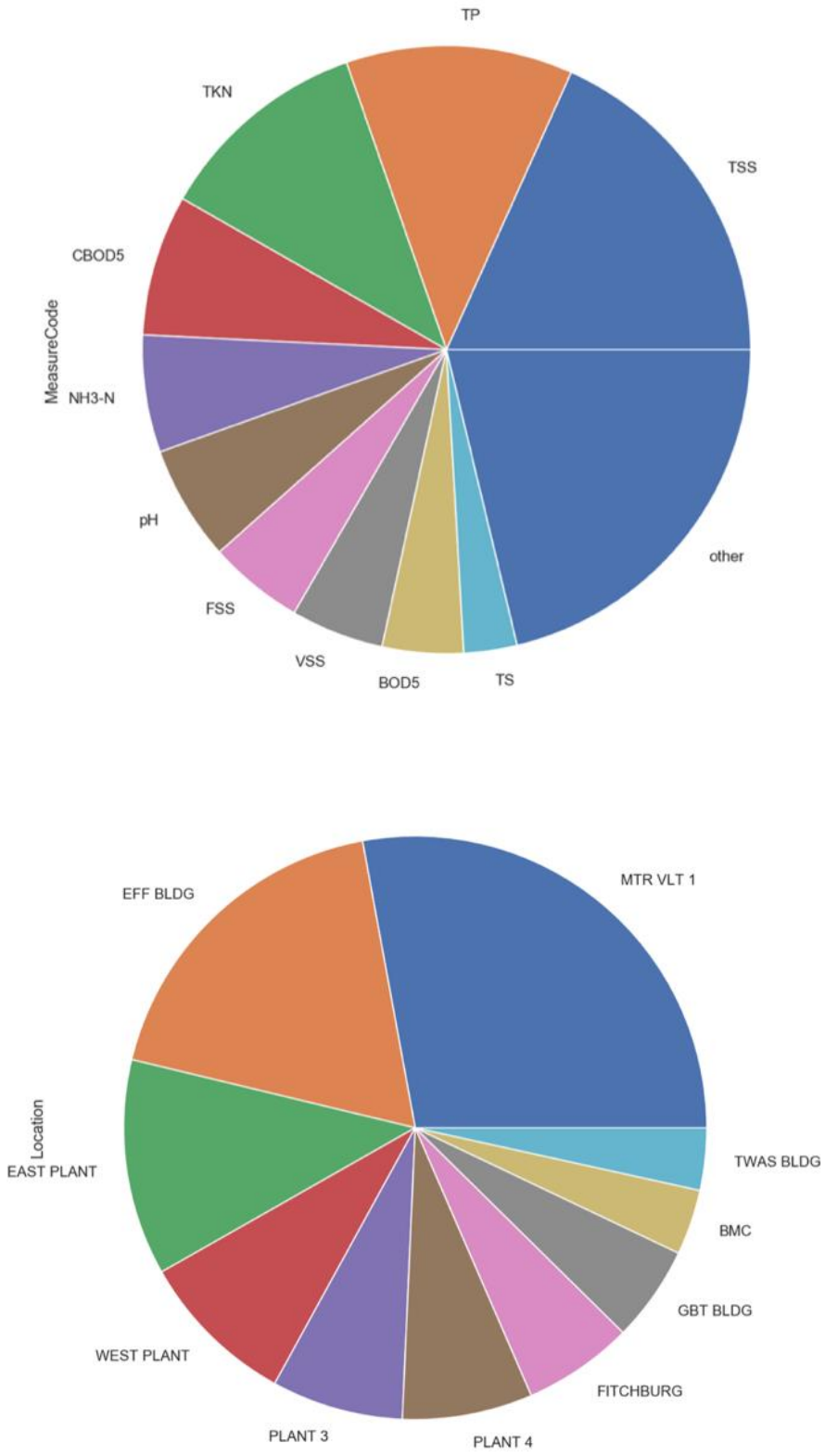


Figure 3.4 Column name is 'MeasureCode', and 'LocationCode' contains various parameters.

The second dataset is the daily dataset contains 8,859 rows and 18 columns of data, which is only 2.9 MB. The data includes flow rate, temperature, DO, pH, and other influent and effluent parameters. However, this chapter will focus on the first dataset, big data from the plant. Still, some parameters in the second dataset will be integrated to analyze the relationship between parameters.

(2) Data understanding

The data collected needs to be studied and understood. The Madison Metropolitan Sewerage District 50-Year Master Plan was reviewed to research the background, goals, and WWTP processes.

(3) Data preparation

In this stage, the data was selected and the form was determined. Data cleaning, visualizing, transforming, and obtaining feature selection and extraction are part of this stage. The data are now available in a form that is compatible with a modeling technique, which will be introduced in the next chapter.

(4) Data mining or data preprocessing.

Various statistical methods were employed to extract knowledge from the preprocessed data. Visualization and statistical analysis, such as Pearson's correlation coefficient, normal distribution, box-plot, and hypothesis testing, were implemented. Data pattern identification, statistical analytics, and normalization are performed in this step.

(5) Evaluation

The results from the previous step were interpreted. The impact of new knowledge was evaluated to determine whether the goals have been met.

3.3.1 Data Preprocessing

Data preprocessing is how the data is transformed or encoded to state that a computer can easily parse it. Data preprocessing helps the computer to understand data. Han et al. (2012) summarized the following steps involved in data preprocessing (Figure 3.5):

(1) Data cleaning

Data cleaning includes many tasks such as filling in missing values, smoothing noisy data, identifying or removing outliers, and correcting inconsistencies. Unclean data can cause uncertainty for the mining process, resulting in inaccurate output. Thus, the data cleaning routine is one of the most important techniques of data preprocessing.

(2) Data integration

The integration of multiple databases or data integration is often required in data mining processes. Data integration is the combining of data from various data stores. Thorough integration helps to reduce and avoid redundancies and inconsistencies in the dataset. Data integration can help increase the precision and speed of the mining process. The challenge of data integration is how we can match schema and objects from different sources. It is the essence of the entity identification problem. The techniques involve correlation tests, duplication recognition, and detection of data value conflicts.

(3) Data transformation

The data will be transformed or consolidated, so the result after the analytics process will be more efficient, and the patterns may be simpler and easier to understand. Strategies for data transformation include smoothing, aggregation, normalization, feature construction, and so on.

(4) Data reduction

Data reduction reduces the size of the dataset that is much smaller in volume and still carefully maintains the integrity of the original data. Therefore, the valid data reduction will produce the same or almost the same analytical consequences.

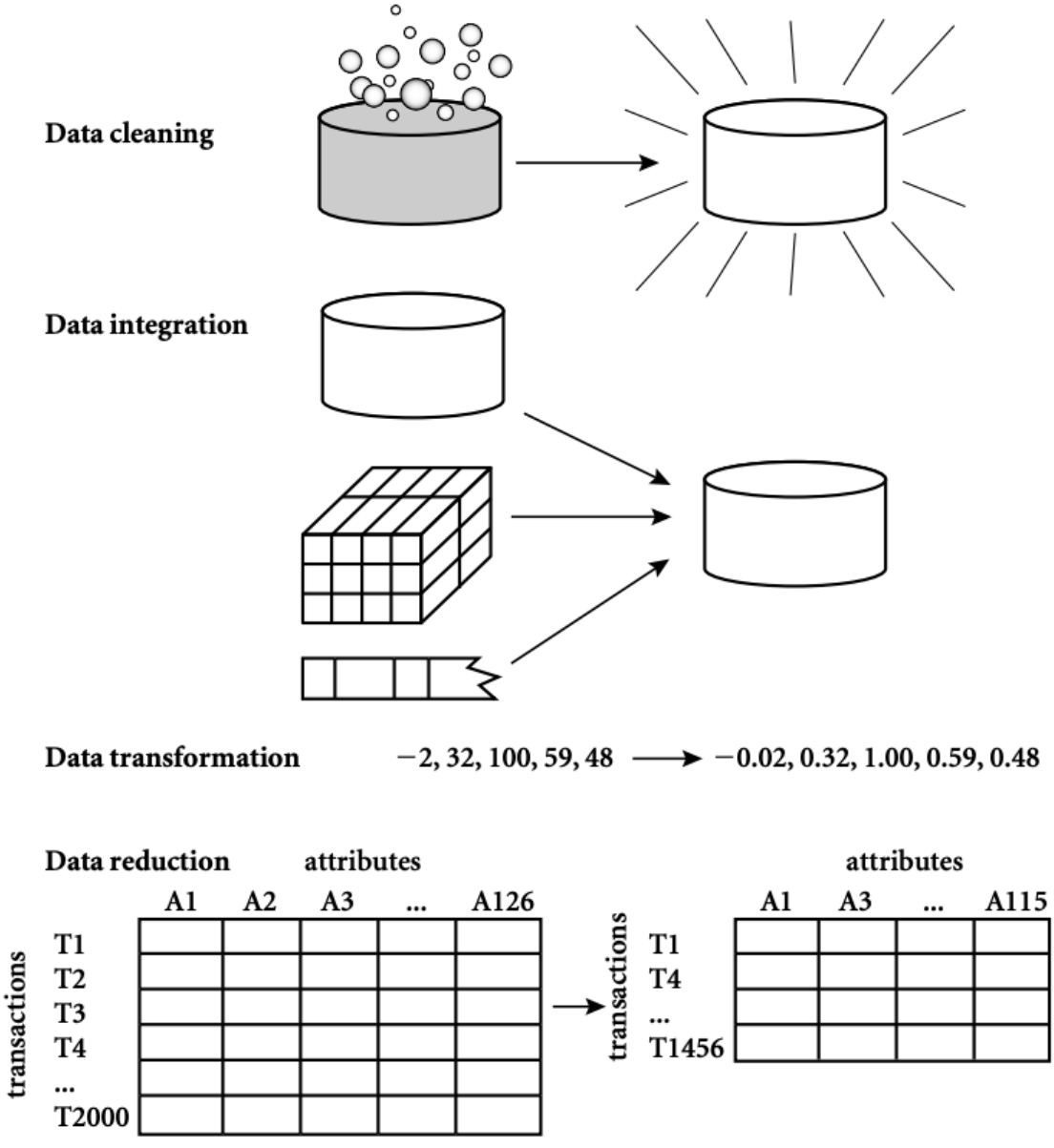


Figure 3.5 Forms of data preprocessing. (Source: Han et al., 2012).

3.3.2 Statistical Analysis

Correlation Coefficient for Numeric Data

For numeric attributes, the correlation between two attributes, A and B, can be evaluated by computing the correlation coefficient. The equation shows below (Han et al., 2011):

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B})}{n\sigma_A\sigma_B} \quad 3.1$$

where n is the number of tuples, a_i and b_i are the respective values of A and B in tuple I, $\overline{A}$ and $\overline{B}$ are the respective mean values of A and B, σ_A and σ_B are the respective standard deviations of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product. Note that $-1 \leq r_{A,B} \leq +1$.

If $r_{A,B}$ is greater than 0, then A and B are positively correlated, meaning that the values of A increase as the values of B increase. The higher the value, the stronger the correlation. Therefore, a higher value may indicate that A (or B) may be removed as a redundancy. If the resulting value equals 0, then A and B are independent, and there is no correlation between them. If the resulting value is less than 0, then A and B are negatively correlated, where the values of one attribute increase as the values of the other attribute decrease. This means that each attribute discourages the other.

Normalization

In statistical analysis, D'Agostino's K2 test measures a goodness-of-fit measure of departure from normality. This test aims to determine whether or not the data sample comes from a normal distribution. The test is from the transformations of the example Kurtosis and Skewness and has power against the alternatives that the distribution is skewed or kurtic (D'agostino et al., 1990).

In the below equation, x_i denotes a sample of n observations, g_1 and g_2 are the sample skewness and Kurtosis, m_j's are the j^{th} sample central moments, and $\bar{x}$ is the sample mean.

The sample skewness and Kurtosis are defined as (D'agostino et al., 1990):

$$g_1 = \frac{m_3}{m_2^{3/2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{3/2}} \quad 3.2$$

$$g_2 = \frac{m_4}{m_2^2} - 3 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^2} - 3 \quad 3.3$$

These quantities consistently estimate the theoretical Skewness and Kurtosis of the distribution, respectively.

Hypothesis Testing

Hypothesis testing in statistics is a method of testing the results to see if there is a meaning in a dataset. Statisticians have developed a way of drawing inferences from samples or finding through hypothesis testing. It can help to interpret data, making decisions, finding errors in results. Hypothesis testing aims to determine the likelihood that a population parameter is likely to be true.

Below are the four steps of hypothesis testing:

(1) Figure out your null hypothesis,

(2) State your null hypothesis,

(3) Choose what kind of test you need to perform, and

(4) Either support or reject the null hypothesis.

3.4 Results and Discussions

3.4.1 Understanding of Data

Background and goals

Madison Metropolitan Sewerage District (MMSD) is a municipal corporation created to collect and treat wastewater from the Madison metropolitan area. MMSD provides service to 43 municipal customers. The service area covers 177 square miles (458 km^2) and serves a current population of nearly 330,000 people. MMSD owns and operates the Nine Springs WWTP (Figure 3.6) (McGowan & Wang, 2008). The Nine Springs WWTP averagely treats 41 million gallons of wastewater per day (155,000 m^3/day). The main objective of the plant is to provide exceptional service at a reasonable cost to customers while considering an appropriate balance between environmental, social, and economic impacts. This study analyzes data generated in the Nine Springs WWTP and finds insights and patterns to optimize WWTP operation.

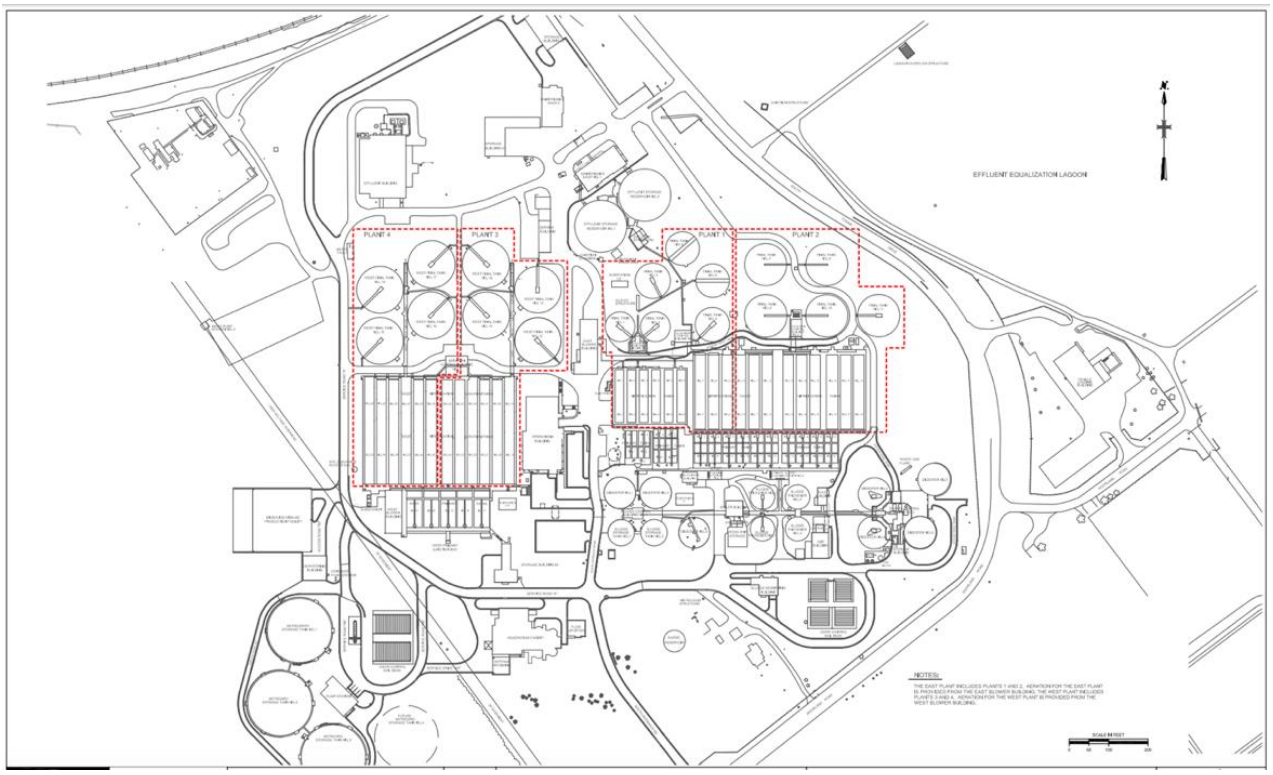


Figure 3.6 General layout of Nine Springs WWTP. (Source: McGowan & Wang, 2008)

The liquid treatment processes

The liquid treatment processes at the Nine Springs WWTP include preliminary treatment, primary clarification, nitrifying activated sludge treatment incorporating biological phosphorus removal, ultraviolet disinfection, excess flow storage, and effluent pumping. Figure 3.7 shows the schematic of the liquid treatment process at the Nine Springs WWTP (McGowan & Wang, 2008).

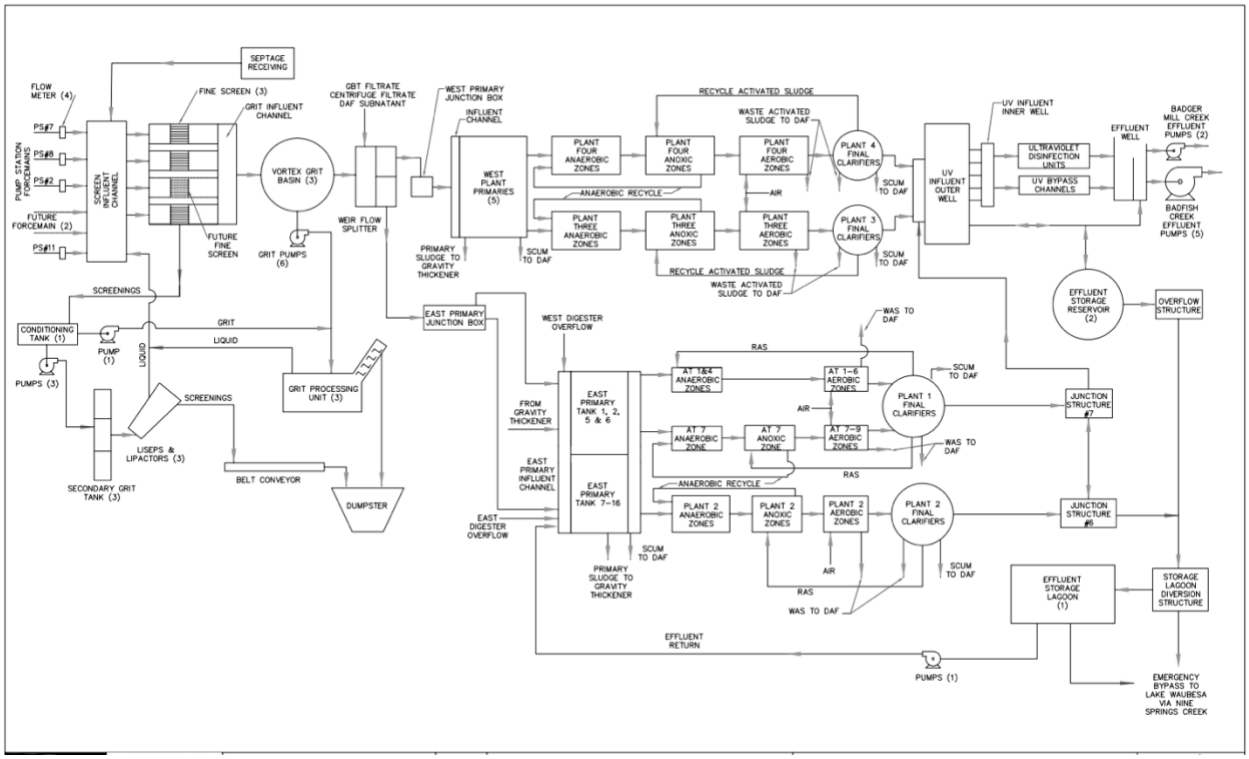


Figure 3.7 The schematic of the liquid treatment process at the Nine Springs WWTP.

(Source: McGowan & Wang, 2008)

The liquid treatment facilities are described below (McGowan & Wang, 2008):

- Headworks

  The headworks consist of influent flow measurement, fine screening, grit removal by vortex grit basins, and a weir flow splitting structure distributing flows to the East and West Complexes. It also consists of screenings and grit processing equipment, the plant water system, and the septage receiving facility. Wastewater enters the headwork facility through influent force mains. After the screening process, the flow continues to vortex grit basins. A sluice trough conveys screenings to screenings processing units. Grit from the vortex grit basins is pumped to grit processing units. Processed screenings and grit are conveyed to roll-off containers by a reversible belt conveyor. The flow exiting the grit basins enters the flow

splitting structure and is distributed to the East and West Complexes through weir troughs with manual stop plates.

- Flow Splitter

  The existing flow splitter splits screened and degritted plant flow between the East and West Complexes using fixed-weir flow-splitting structures.

- Primary Settling Facilities

  There are 14 primary clarifiers in the East Complex and five primary clarifiers in the West Complex. All clarifiers are rectangular units with chain and flight sludge removal mechanisms. Settled primary sludge is pumped to gravity thickeners for thickening before being digested.

- Aeration Basins

  Biological treatment of the primary effluent occurs in the aeration basins. There are 18 aeration basins in the East Complex and 12 in the West Complex. The aeration basins are configured such that each group of three aeration basins functions as one "folded" treatment unit. Thus, there are six treatment units in the East Complex and four treatment units in the West Complex. Aeration tank effluent proceeds into the secondary clarifiers for settling. The existing secondary treatment facility is an enhanced biological phosphorus removal (EBPR) system with two process configurations being utilized – The University of Cape Town (UCT) Variation process, which is used for the majority of the plant, and the anaerobic/aerobic (A/O) process.

  The UCT process consists of anaerobic, anoxic, and aerobic zones. Influent wastewater enters the anaerobic zone and is combined with recycling from the anoxic zone. Mixed liquor flows into the anoxic area created by pumping return activated sludge (RAS) from the final clarifiers. The mixed liquor then proceeds into the aerobic zone for further treatment.

The A/O process is utilized in two of the three treatment units of Plant 1. In the A/O process, the anoxic zone is eliminated and RAS is combined with the influent wastewater in the anaerobic area. Following the anaerobic zone, the mixed liquor flows to the aerobic zone.

- Secondary Clarification Facilities

  Effluent from the aeration tanks flows to secondary clarifiers for settling. There are 11 secondary clarifiers in the East Complex and 8 in the West Complex. The effluent of the secondary clarifiers flows to UV disinfection facilities before being discharged. The RAS is pumped to aeration tanks while waste activated sludge (WAS) and scum are pumped to dissolved air floatation (DAF) thickeners for thickening before being digested.

- UV Disinfection Facilities

  UV disinfection facilities disinfect the effluent from the secondary clarifiers. The existing UV disinfection system is an open channel, low-pressure mercury vapor type. There are seven channels, five of which are installed with UV disinfection equipment, one is reserved for future equipment, and the seventh channel is used as a bypass channel when the UV system is out of service. Each of the five channels has two UV banks in series. Typically, two to four channels are in service with one bank of lamps operational. During peak flow rates, additional UV channels are added to meet the flow demands. Channels are brought online and taken offline by automatically controlled motorized gates installed on the inlet of each channel.

- Plant Effluent Pumping Facilities

  The existing effluent pumping facilities were constructed during the plant's Seventh Addition. The plant effluent is pumped to Badfish Creek through a 54" force main of 5 miles and Badger Mill Creek through a 20" force main of 10 miles. The Badfish Creek effluent pumps consist of five horizontal split case centrifugal pumps, each with an 800 hp, 880 rpm motor. Three pumps

are outfitted with 25.94" (65.9 cm) diameter impellers, and two are equipped with impellers trimmed to 24" (61.0 cm) to save energy when lower flow rates are practicable. The Badger Mill Creek effluent pumps include two centrifugal pumps, each with a 200 hp, variable speed motor. Each pump has a capacity of 2,000 gallons per minute (gpm) (7.57 m^3/min) at a total dynamic head of 190 feet (57.9 m).

- Effluent Storage Facilities

  The plant has two effluent storage tanks and an effluent storage lagoon for plant effluent storage. The disinfected effluent beyond effluent pumping capacity and up to an estimated flow rate of 115 mgd (435,000 m^3/day) overflows to effluent storage reservoirs. The effluent storage reservoirs, in turn, overflow to the effluent storage lagoon when their maximum storage capacities are reached. Flows over 115 million gallons per day (mgd) (435,300 m^3/day) (estimated) receive secondary treatment and are diverted to the effluent equalization facilities. This estimated flow rate is based on a flow split at the flow splitter of 45% to the east side and 55% to the west side of the plant. At a total flow of 115 mgd, the East Complex flow would be 52 mgd (197,000 m^3/day), which is the flow rate from the east side final clarifiers at which bypassing the secondary effluent was observed previously. The effluent equalization facilities (storage lagoons) have a nominal volume of 50 million gallons (190,000 m^3). When this volume is exceeded, an overflow structure diverts additional flows to the ditch on the north side of the lagoons. Flow in the ditch discharges to Nine Springs Creek, which in turn discharges to Lake Waubesa. Discharges to the effluent equalization facilities are pumped back to the secondary process when the plant peak flow subsides. Since the effluent storage lagoons are open to the atmosphere, the effluent storage volume is reduced by 1.3 million gallons (4,900 m^3) for each inch of precipitation.

3.4.2 Data Preparation

The types of data collected are in the SQL Database and the Excel file. Python Jupyter Notebook, which is open-source software containing live code, equations, and visualization, was used in this study. The Jupyter™ is registered with the U.S. Patent & Trademark Office. The program applications include data cleaning, data visualization, statistical modeling, machine learning, etc. The program was used to analyze, select, preprocess, visualize, and transform the large-size data into the appropriate dataset to develop a prediction model. After processing the first dataset, data were selected only the data designated as 'NS,' which means 'Nine Springs WWTP' from the "SiteCode" columns. The influent parameters will be selected from 'MTR VLT1', the influent meter vault where the wastewater has entered the plant. The effluent parameters will be chosen from 'EFF BLDG,' which is the effluent building where the treated water is sent. Table 3.1 shows the number of rows in each value in each column.

Table 3.1 The selection of parameters from 'SiteCode' and 'LocationCode'.

```
In [37]: Site = df['SiteCode'].value_counts()
         Site

Out[37]: NS            707303
         UC            190582
         SRW            38495
         BFC            24013
         SEP            10030
         UYRW            9719
         IP              8207
         RRW             3499
         NSC             3373
         WMRS            1604
         BELLEVILLE      1458
         VER             1222
         BROOKLYN         858
         Name: SiteCode, dtype: int64
```

```
In [39]: Location = df['LocationCode'].value_counts()
         Location

Out[39]: MTR VLT 1     167531
         EFF BLDG      110187
         EAST PLANT     71949
         WEST PLANT     52704
         PLANT 3        43848
         PLANT 4        43397
         FITCHBURG      36668
         GBT BLDG       31897
         BMC            21720
         TWAS BLDG      20693
         Name: LocationCode, dtype: int64
```

After selecting the locations, the data is separated into the influent and effluent table. There are 167,531 rows × 13 columns for influent in Table 3.2 and 110,001 rows × 13 columns for effluent in Table 3.3. However, there are too many columns and parameters in each table, so the next step is to determine the essential parameters and the meaningful columns to find out insight from this vast dataset.

Table 3.2 The influent table from 'MTR VLT' location.

```
In [4]: Influent = df.loc[(df['DescriptionCode']=='INFLUENT')& (df['LocationCode']=='MTR VLT 1')]
        Influent
```

Out[4]:

TypeMethodID	ResultDate	OriginalResult	Result	AnalysisDate	IsBadData	SiteCode	LocationCode	DescriptionCode	LabReferenceName	MeasureCode
20.0	1997-01-09	34.00	34.00	1997-01-10 00:00:00.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	FSS
20.0	1997-01-10	25.00	25.00	1997-01-11 00:00:00.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	FSS
20.0	1997-01-11	29.00	29.00	1997-01-12 00:00:00.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	FSS
20.0	1997-01-12	29.00	29.00	1997-01-13 00:00:00.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	FSS
20.0	1997-01-13	22.00	22.00	1997-01-14 00:00:00.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	FSS
...	...	...	...	...	...	...	...	...	...	...
48.0	2019-01-02	5.00	5.00	2019-01-07 11:21:01.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	TP
48.0	2019-01-04	5.42	5.42	2019-01-07 11:21:01.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	TP
112.0	2019-01-05	206.00	206.00	2019-01-08 06:56:21.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	TSS
112.0	2019-01-05	188.00	188.00	2019-01-08 06:56:21.000	0	NS	MTR VLT 1	INFLUENT	PLT RAW	VSS
112.0	2019-01-05	201.00	201.00	2019-01-08 06:56:21.000	0	NS	MTR VLT 1	INFLUENT	UC PS-18	TSS

Table 3.4 shows the parameters in 'MeasureCode' columns. Thus, we will extract, split, and create a meaningful table for the significant parameters in influent and effluent in Nine Springs WWTP.

Table 3.3 The effluent table from 'EFF BLDG' location.

```
In [5]: Effluent = df.loc[(df['DescriptionCode']=='EFFLUENT')& (df['LocationCode']=='EFF BLDG')]
        Effluent
```

Out[5]:

TypeMethodID	ResultDate	OriginalResult	Result	AnalysisDate	IsBadData	SiteCode	LocationCode	DescriptionCode	LabReferenceName	MeasureCode
1.0	1996-01-01 00:00:00	5.00	5.00	1996-01-07 00:00:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	BOD5
1.0	1996-01-02 00:00:00	5.00	5.00	1996-01-08 00:00:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	BOD5
1.0	1996-01-03 00:00:00	6.00	6.00	1996-01-09 00:00:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	BOD5
1.0	1996-01-04 00:00:00	5.00	5.00	1996-01-10 00:00:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	BOD5
1.0	1996-01-05 00:00:00	5.00	5.00	1996-01-11 00:00:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	BOD5
...	...	...	...	...	...	...	...	...	...	...
48.0	2019-01-04 00:00:00	0.25	0.25	2019-01-07 11:21:01.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	TP
112.0	2019-01-05 00:00:00	4.20	4.20	2019-01-08 06:56:21.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	TSS
112.0	2019-01-05 00:00:00	4.00	4.00	2019-01-08 06:56:21.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	VSS
90.0	2019-01-05 10:04:00	7.30	7.30	2019-01-05 10:07:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	pH
90.0	2019-01-06 15:57:00	7.30	7.30	2019-01-06 16:05:00.000	0	NS	EFF BLDG	EFFLUENT	PLT CE	pH

Table 3.4 Number of parameters in 'MeasureCode'.

```
In [6]: Influent['MeasureCode'].value_counts()
Out[6]: TSS        26045
        CBOD5      21757
        TKN        21750
        TP         21750
        pH         20356
        VSS        17607
        FSS        17531
        NH3-N       8395
        BOD5        8395
        Cl          1925
        Hg           277
        Cr           277
        Cd           276
        Ni           276
        Pb           276
        Zn           276
        Cu           276
        CN-T          47
        OP-DRA        32
        COND           7
        Name: MeasureCode, dtype: int64
```

```
In [7]: Effluent['MeasureCode'].value_counts()
Out[7]: TSS        15198
        NH3-N      15059
        TP         15057
        BOD5       15057
        pH          9126
        FSS         8404
        VSS         8404
        TKN         8402
        Cl          5196
        NO3-N       2519
        SO4         2437
        FCOLI       1263
        OP-DRA       635
        Cr           279
        NO3+NO2      279
        Cu           279
        Pb           278
        Zn           278
        Ni           278
```

Table 3.5 shows the dataset, which is easier to understand and ready for the following process.

Table 3.5 The clean dataset from the Nine Springs WWTP big data.

	ResultDate	Influent_TSS	Effluent_TSS	Influent_TP	Effluent_TP	Influent_TKN	Effluent_TKN	Influent_NH3N	Effluent_NH3N	Influent_BOD5	Effluent_BOD5
0	1997-01-02	242.0	7.0	6.42	2.21	31.4	1.45	19.9	0.18	196.0	4.0
1	1997-01-02	242.0	7.0	6.42	2.21	31.4	1.45	19.9	0.18	196.0	4.0
2	1997-01-02	242.0	7.0	6.42	2.21	31.4	1.45	19.9	0.18	196.0	4.0
3	1997-01-02	242.0	7.0	6.42	2.21	31.4	1.45	19.9	0.18	196.0	4.0
4	1997-01-02	242.0	7.0	6.42	2.21	31.4	1.45	19.9	0.18	196.0	4.0
...	...	...	...	...	...	...	...	...	...	...	...
927	2019-01-02	320.0	4.7	5.00	0.26	41.8	2.08	27.7	0.11	257.0	6.3
928	2019-01-02	178.0	4.7	5.00	0.26	41.8	2.08	27.7	0.11	257.0	6.3
929	2019-01-02	210.0	4.7	5.00	0.26	41.8	2.08	27.7	0.11	257.0	6.3
930	2019-01-02	197.0	4.7	5.00	0.26	41.8	2.08	27.7	0.11	257.0	6.3
931	2019-01-02	252.0	4.7	5.00	0.26	41.8	2.08	27.7	0.11	257.0	6.3

Data visualization

Data visualization means the graphic representation of data (Few, 2014). The relationships between influent and effluent in each parameter are shown in Figures 3.8 to 3.12. The selected parameters include TSS, TP, TKN, NH_3N, and BOD_5, which are the essential parameters affecting wastewater treatment quality.

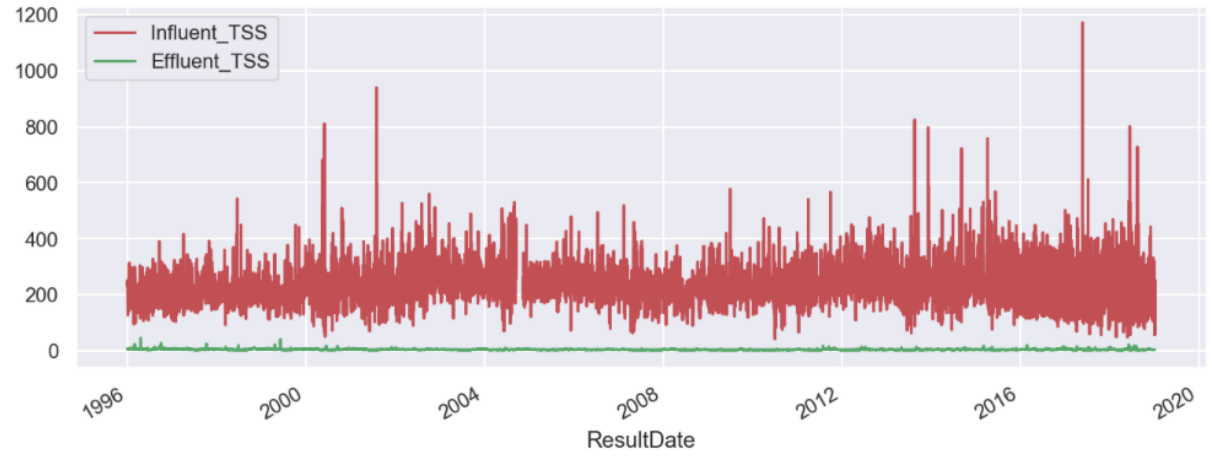


Figure 3.8 The relationship between influent and effluent in TSS.

Figure 3.9 The relationship between influent and effluent in TP.

Figure 3.10 The relationship between influent and effluent in TKN.

Figure 3.11 The relationship between influent and effluent in NH_3N.

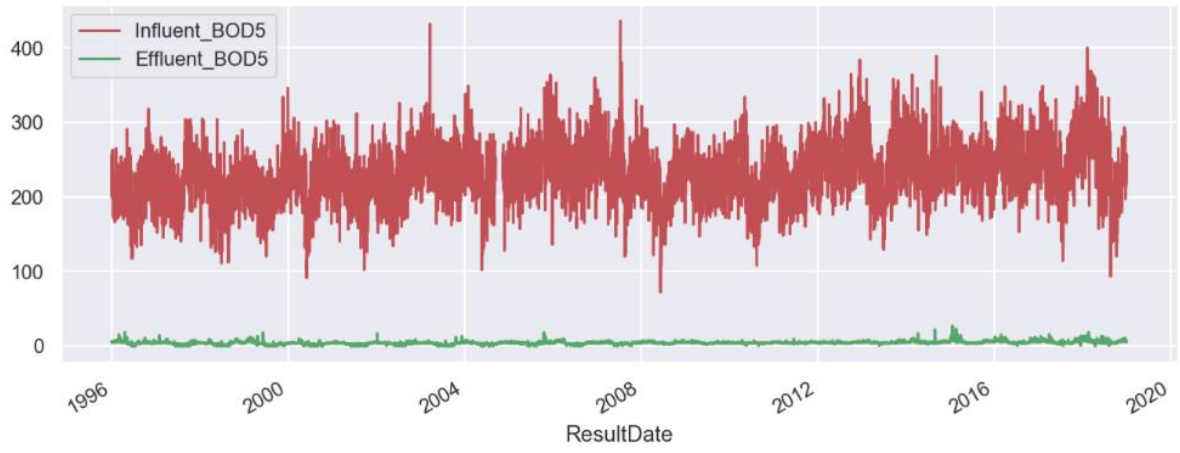


Figure 3.12 The relationship between influent and effluent in BOD_5.

In the second dataset, the daily data was selected in the same location, which is from the Nine Springs WWTP Influent Meter Vault and Effluent Building for influent and effluent data, respectively.

3.4.3 Data Preprocessing

Statistical Analysis

In the Python Jupyter notebook, a STAT module can help summarize data using the 'describe a function,' describe(). The describe() function is used to generate descriptive statistics that summarize the central tendency and dispersion of the numerical values in the dataset. The values of parameters - mean, standard deviation, percentile, and the interquartile range of the data - are shown in Table 3.6.

Table 3.6 The table of descriptive statistics.

	count	mean	std	min	25%	50%	75%	max
Influent_TSS	3713349.0	231.842439	52.453469	43.00	200.00	225.00	258.00	1170.00
Effluent_TSS	3713349.0	4.818173	1.627289	0.00	3.90	4.60	5.50	46.00
Influent_TP	3713349.0	5.955506	1.045251	1.22	5.25	5.91	6.58	17.00
Effluent_TP	3713349.0	0.562397	0.859708	0.00	0.23	0.31	0.43	5.78
Influent_TKN	3713349.0	35.922470	6.945648	4.43	31.20	35.40	40.10	95.30
Effluent_TKN	3713349.0	1.477263	0.419796	0.00	1.25	1.41	1.64	19.30
Influent_NH3N	3713349.0	22.507716	4.055141	0.00	19.90	22.10	24.90	40.10
Effluent_NH3N	3713349.0	0.167732	0.286970	0.00	0.05	0.08	0.18	16.90
Influent_BOD5	3713349.0	230.064854	37.164753	72.00	207.00	230.00	252.00	436.00
Effluent_BOD5	3713349.0	3.654993	1.709473	0.00	2.80	3.60	4.40	27.00

Correlation Coefficient

The relationship between factors can be analyzed by using the correlation coefficient calculation. Table 3.7 shows the values of the correlation coefficient between all parameters. Table 3.5 can be used to determine the strength of the relationship of each parameter and select input and output parameters for our developing model in the next chapter.

Table 3.7 Correlation coefficient between parameters.

	Influent_TSS	Effluent_TSS	Influent_TP	Effluent_TP
Influent_TSS	1.000000	0.008635	0.186263	-0.118263
Effluent_TSS	0.008635	1.000000	0.179999	0.362095
Influent_TP	0.186263	0.179999	1.000000	0.129379
Effluent_TP	-0.118263	0.362095	0.129379	1.000000
Influent_TKN	0.202300	-0.082926	0.188097	-0.250782
Effluent_TKN	0.025458	0.322771	0.081531	0.209102
Influent_NH3N	0.201944	-0.038195	0.160931	-0.208171
Effluent_NH3N	0.038189	0.146196	0.073981	0.046628
Influent_BOD5	0.345516	0.020911	0.382486	-0.133069
Effluent_BOD5	0.019187	0.443104	0.095095	0.080433

	Influent_TKN	Effluent_TKN	Influent_NH3N	Effluent_NH3N
Influent_TSS	0.202300	0.025458	0.201944	0.038189
Effluent_TSS	-0.082926	0.322771	-0.038195	0.146196
Influent_TP	0.188097	0.081531	0.160931	0.073981
Effluent_TP	-0.250782	0.209102	-0.208171	0.046628
Influent_TKN	1.000000	0.206942	0.718030	0.109953
Effluent_TKN	0.206942	1.000000	0.278937	0.726848
Influent_NH3N	0.718030	0.278937	1.000000	0.181020
Effluent_NH3N	0.109953	0.726848	0.181020	1.000000
Influent_BOD5	0.522786	0.226732	0.602963	0.170096
Effluent_BOD5	0.341836	0.580882	0.428807	0.449895

	Influent_BOD5	Effluent_BOD5
Influent_TSS	0.345516	0.019187
Effluent_TSS	0.020911	0.443104
Influent_TP	0.382486	0.095095
Effluent_TP	-0.133069	0.080433
Influent_TKN	0.522786	0.341836
Effluent_TKN	0.226732	0.580882
Influent_NH3N	0.602963	0.428807
Effluent_NH3N	0.170096	0.449895
Influent_BOD5	1.000000	0.297437
Effluent_BOD5	0.297437	1.000000

When the correlation coefficient between the two parameters is greater than 0, they are positively correlated. It means that when one value increases, another value also increases. The higher the correlation coefficient value, the stronger the correlation. Table 3.8 shows the heatmap colors of correlation. The lighter color means that they have a stronger relationship. However, if the color turns black, it shows a negative value, indicating that each attribute discourages another. Effluent BOD_5 has all positive values with higher correlations than other parameters. Therefore, the effluent BOD_5 was used as an output parameter for the first model. After that, other parameters will be used as inputs. Figure 3.13 shows the correlations of effluent BOD_5 to other parameters such as flow rate, pH, temperature, and other effluent parameters.

Table 3.8 Heatmap of the correlation coefficient.

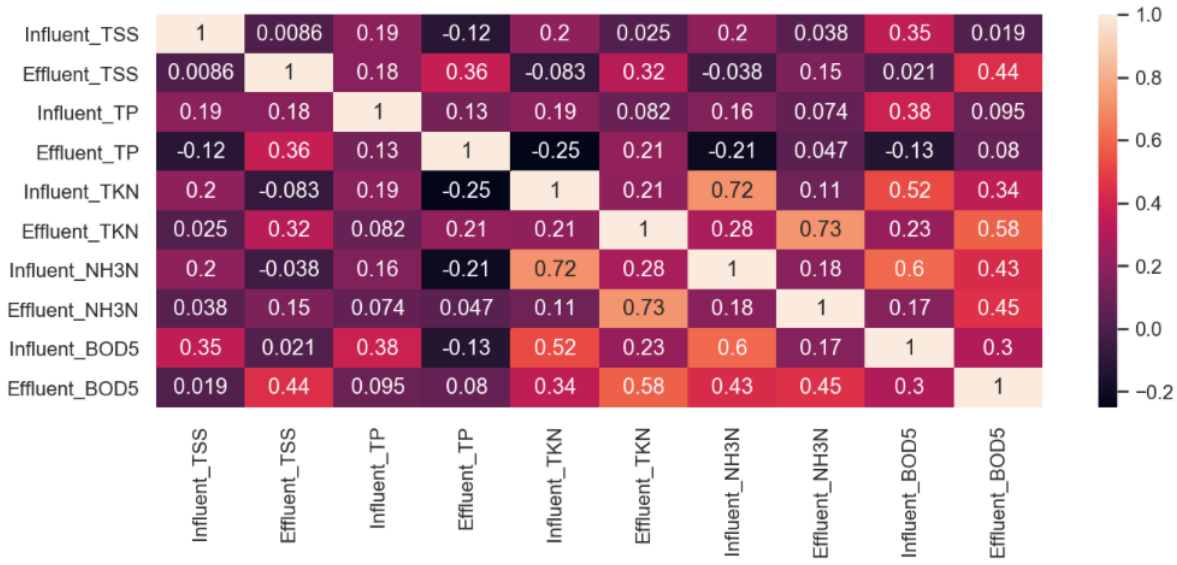


Figure 3.13 Correlation of effluent BOD_5 to other parameters.

Removal of missing value and merging of date and time data

The data preprocessing steps are to: (1) merge date and time into one column and change to DateTime type, (2) convert all data to numeric, (3) remove missing values, and (4) create year, quarter, month, and day features. After removing the missing values, the data contains 3,713,349

measurements collected between January 1996 and January 2019 (Table 3.9). The initial data includes several variables. However, the statistical analysis will focus on a single value: Historical Effluent BOD_5 data.

Table 3.9 Program for the result after removing missing values.

```
In [3]: df['date_time'] = pd.to_datetime(df.index)
        df['Effluent_BOD5'] = pd.to_numeric(df['Effluent_BOD5'], errors='coerce')
        df = df.dropna(subset=['Effluent_BOD5'])
        df['ResultDate']=pd.to_datetime(df['date_time'])
        df['year'] = df['date_time'].apply(lambda x: x.year)
        df['quarter'] = df['date_time'].apply(lambda x: x.quarter)
        df['month'] = df['date_time'].apply(lambda x: x.month)
        df['day'] = df['date_time'].apply(lambda x: x.day)
        df=df.loc[:,['date_time','Effluent_BOD5', 'year','quarter','month','day']]
        df.sort_values('date_time', inplace=True, ascending=True)
        df = df.reset_index(drop=True)

        print('Number of rows and columns after removing missing values:', df.shape)
        print('The time series starts from: ', df.date_time.min())
        print('The time series ends on: ', df.date_time.max())

        Number of rows and columns after removing missing values: (3713349, 6)
        The time series starts from:  1996-01-02 00:00:00
        The time series ends on:  2019-01-02 00:00:00
```

Normalization

a. Statistical Normality Test

Several statistical tests can be applied to quantify whether the data was drawn from a Gaussian distribution. D'Agostino's K^2 statistical test will be implemented in Python Jupyter. The p-value is interpreted as follows:

- $p \leq$ alpha: reject H_0, not normal; and
- $p >$ alpha: fail to reject H_0, normal.

Table 3.10 shows the result of the statistical analysis shows that effluent BOD_5 data reject H_0, which means that the data is not a Gaussian distribution (not normal).

Table 3.10 Program for processing normal distribution test.

```
In [4]: stat, p = stats.normaltest(df.Effluent_BOD5)
        print('Statistics=%.3f, p=%.3f' % (stat, p))
        alpha = 0.05
        if p > alpha:
            print('Data looks Gaussian (fail to reject H0)')
        else:
            print('Data does not look Gaussian (reject H0)')

        Statistics=716318.979, p=0.000
        Data does not look Gaussian (reject H0)
```

Kurtosis and Skewness can also determine if the data distribution departs from the normal distribution (Li, 2019). Kurtosis describes the heaviness of the tails of a distribution. When a kurtosis is close to 0, it is a normal distribution. If the Kurtosis is greater than zero, the distribution has heavier tails. If the Kurtosis is less than zero, then the distribution is light tails. Skewness measures the asymmetry of the distribution. If the Skewness is between -0.5 and 0.5, the data are relatively symmetrical. If the Skewness is between -1 and - 0.5 or between 0.5 and 1, the data are moderately skewed. If the Skewness is less than -1 or greater than 1, the data are highly skewed. Figure 3.14 shows that our Kurtosis is 5.205, meaning the heaviness of the tails of a distribution. The skewness of normal distribution is 0.779, implying the data are moderately skewed.

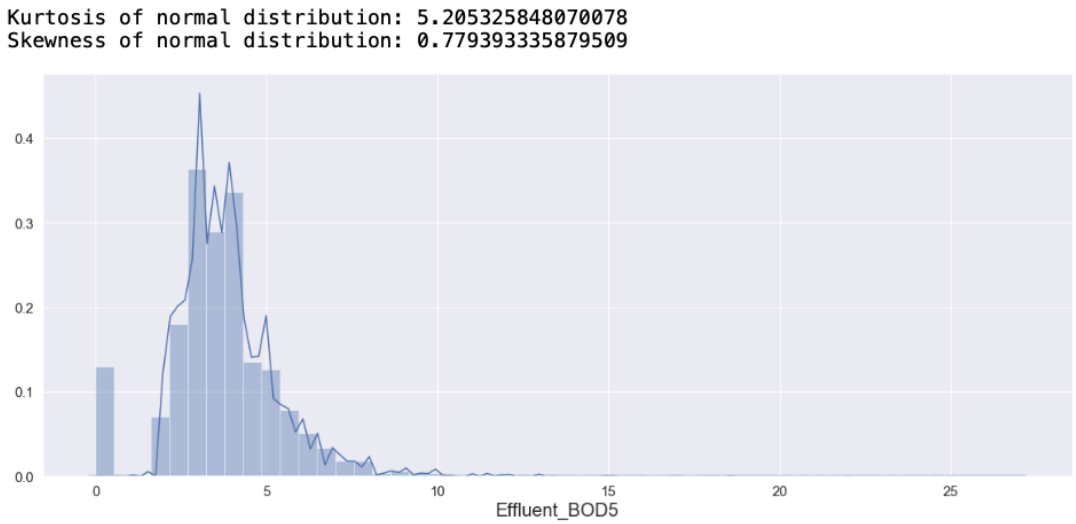


Figure 3.14 Kurtosis and Skewness of normal distribution.

b. Box plot

Box plot is another tool for visualizing data. Figure 3.15 shows the yearly box plot, noticing that the median effluent BOD_5 in 2014 is higher than the other years. The median effluent BOD_5 values were higher in the first and fourth quarters (winter) and the lowest in the third quarter (summer).

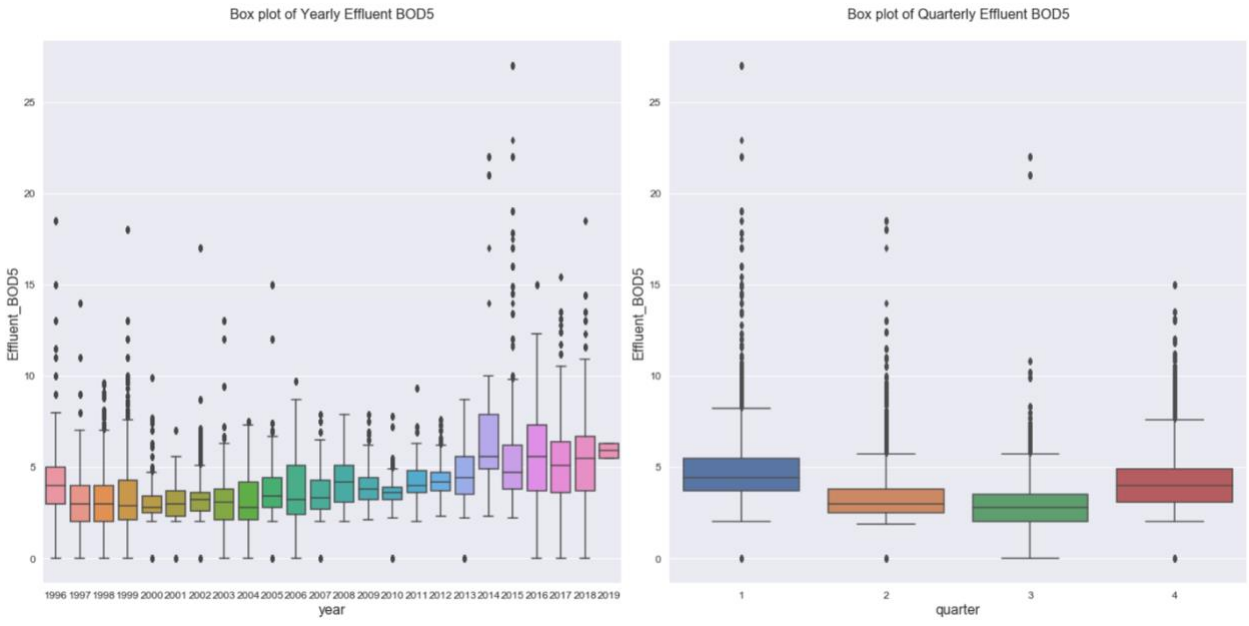


Figure 3.15 Box plot of yearly and quarterly effluent BOD_5.

c. Normal probability distribution

The normal probability plot, Figure 3.16, also shows that the effluent BOD_5 data is far from normally distributed.

	count	mean	std	min	25%	50%	75%	max
Effluent_BOD5	3713349.000	3.655	1.709	0.000	2.800	3.600	4.400	27.000

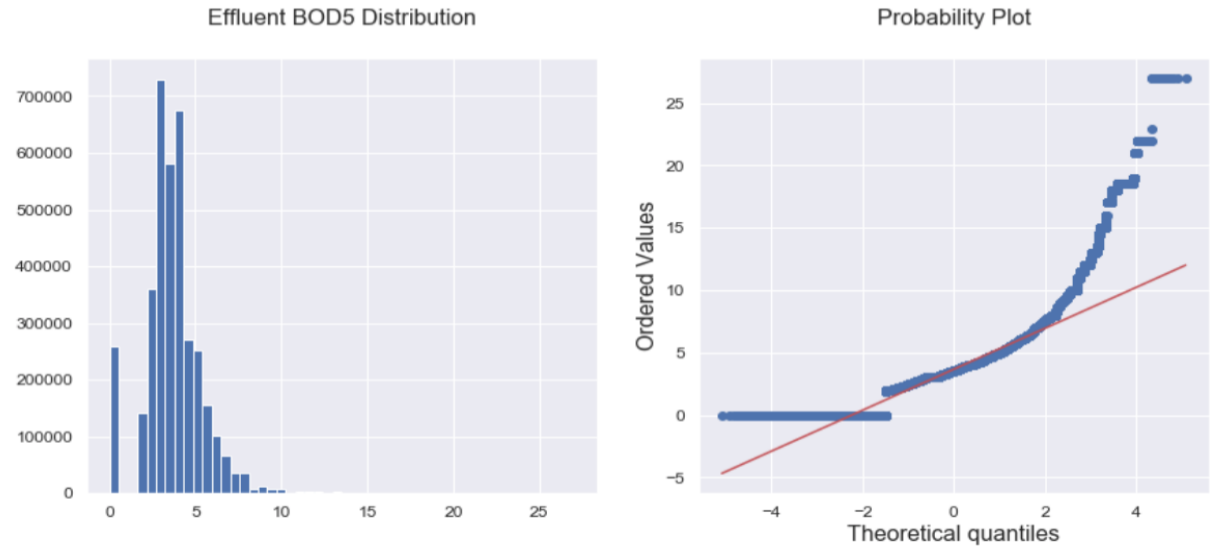


Figure 3.16 Normal probability distribution.

Figure 3.17 represents the means of effluent BOD_5 over days, weeks, months, quarters, and years. The highest year of effluent BOD_5 is in 2014, where data is missing in a certain period. Therefore, the results show that the data is not normal.

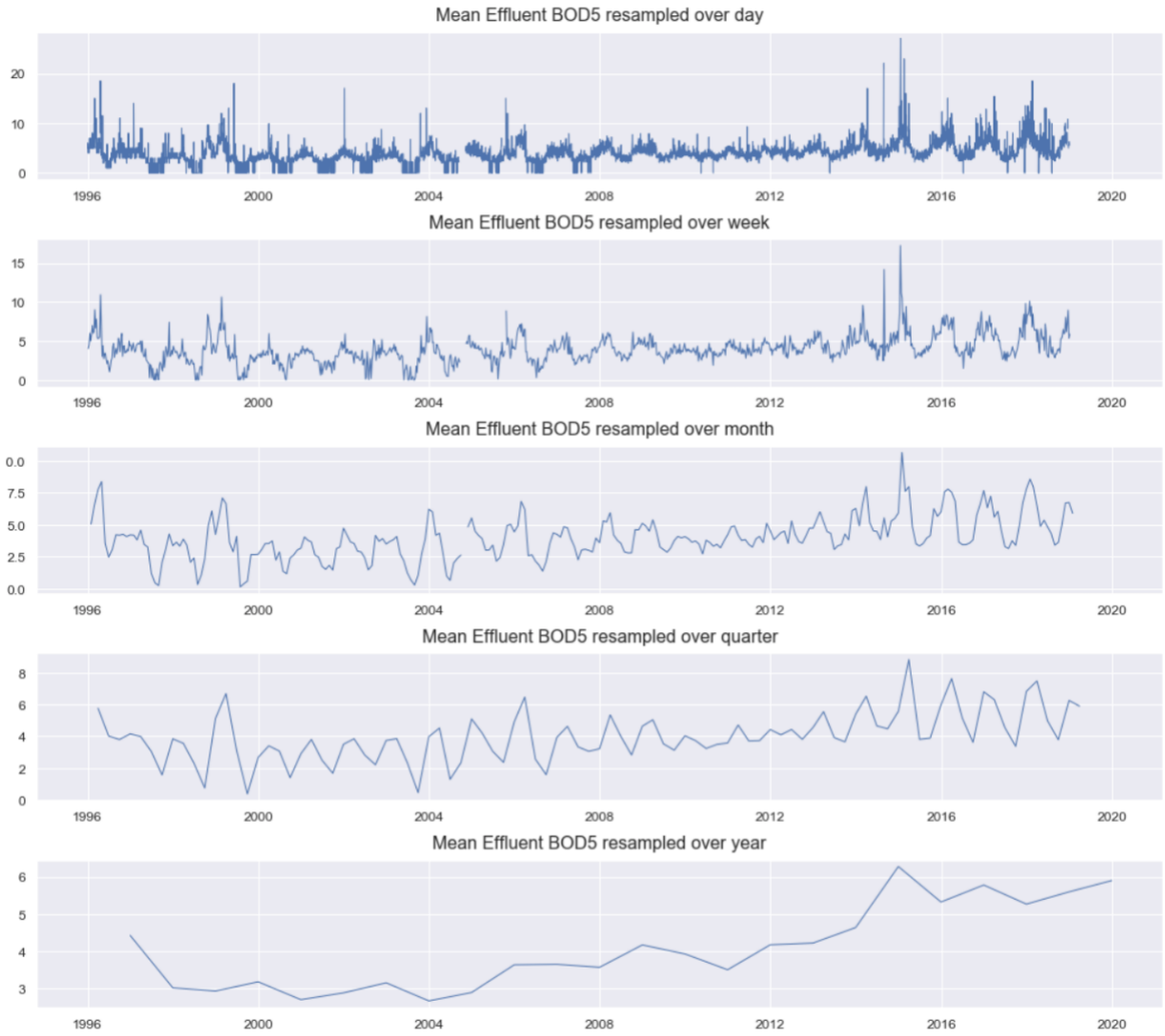


Figure 3.17 The means effluent BOD_5 over a day, week, month, quarter, and year.

Figure 3.18 confirms the previous analysis in Figure 3.17. The highest yearly effluent BOD_5 was in 2014. The lowest quarterly average effluent BOD_5 was in the third quarter. The lowest monthly effluent BOD_5 was in August. The lowest daily average effluent BOD_5 was around the 30th of the month.

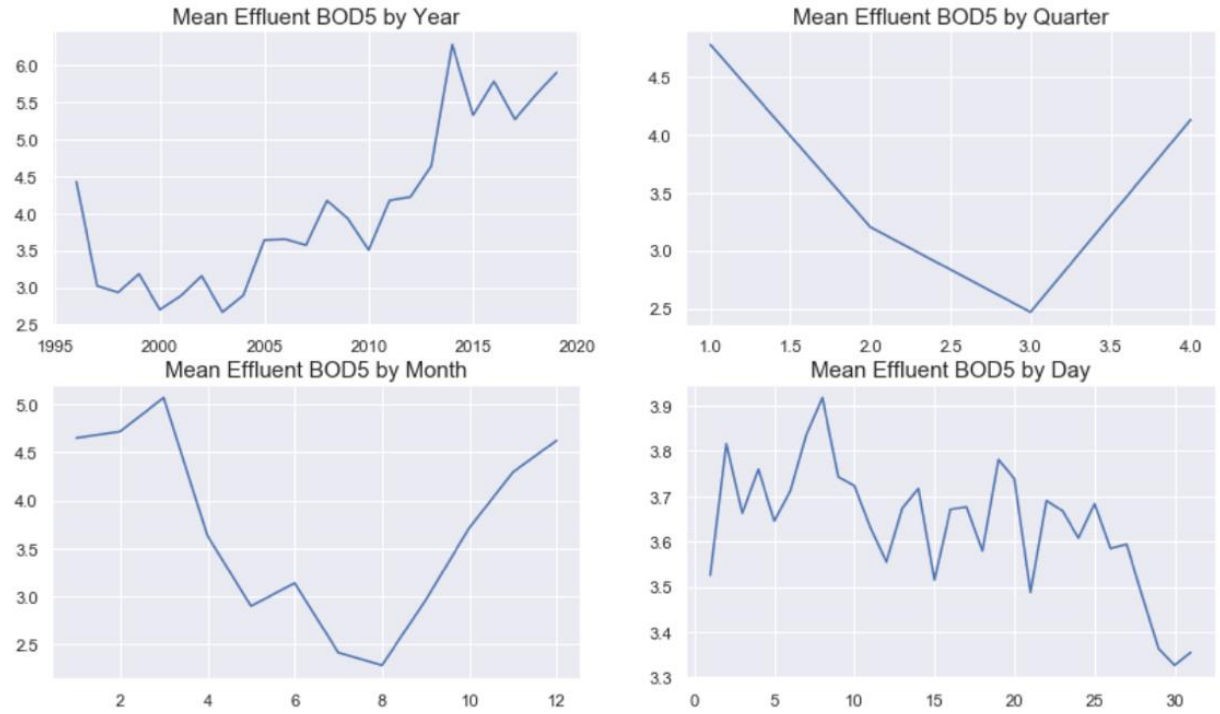


Figure 3.18 The average means effluent BOD_5 by year, quarter, month, and day.

Figure 3.19 shows the patterns of the effluent BOD_5 for each year. As a result, 2004 and 2019 data were removed because the data was missing and 2014 data was removed because the values were too high.

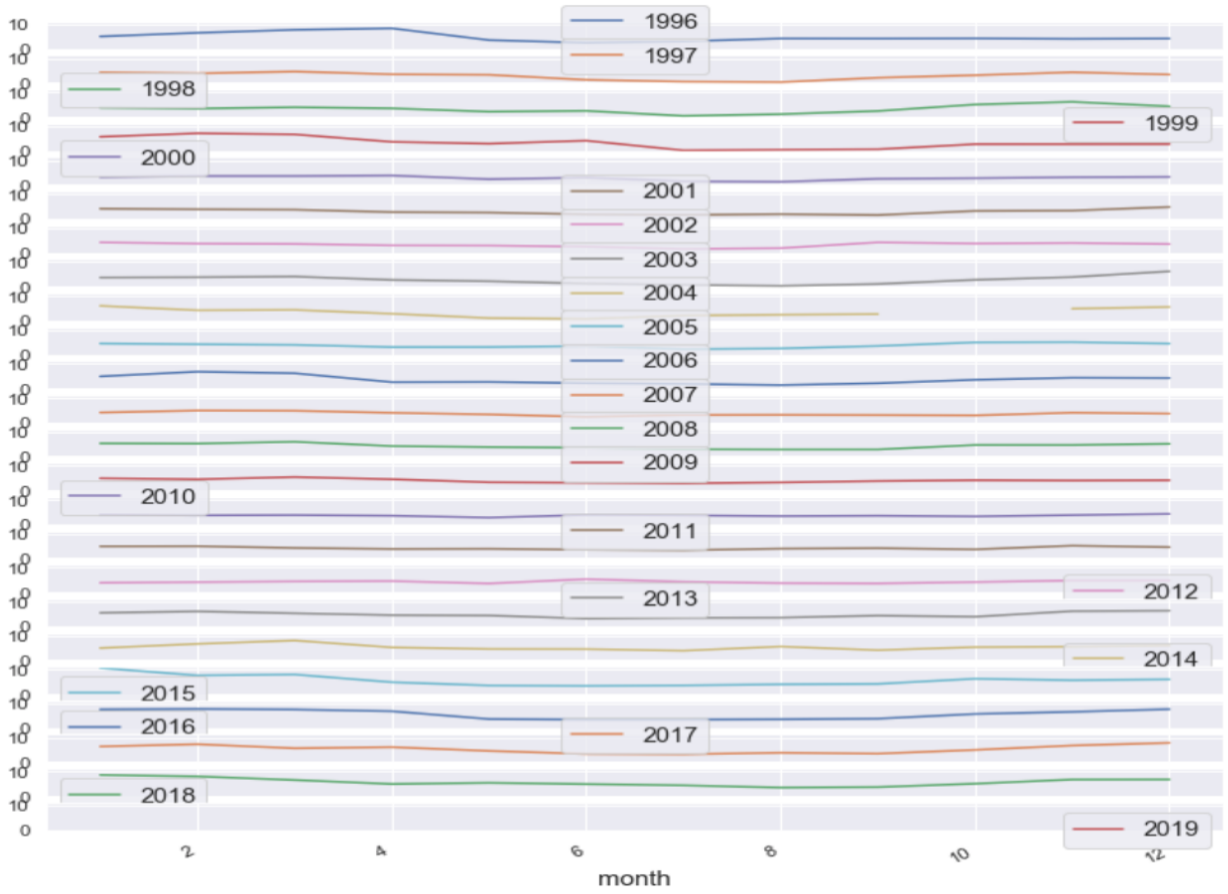


Figure 3.19 The patterns effluent BOD_5 for each year.

Therefore, the data of 2005-2013 or 2015-2018 was selected for a more accurate prediction model. In the next section, the data of 2015-2018 was evaluated for accuracy and trend.

3.4.4 Evaluation

The data of 2015-2018 are further evaluated. In Figure 3.20, the data is plotted to determine the pattern. The box plot in Figure 3.21 displays how close the data is in each year and quarter.

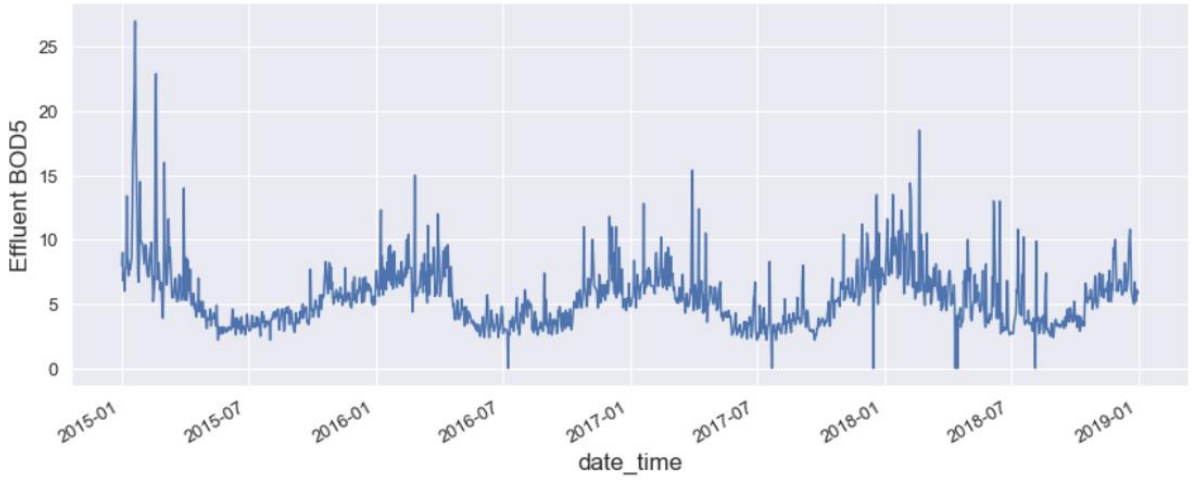


Figure 3.20 Time series plot of effluent BOD_5 from 2015-2018.

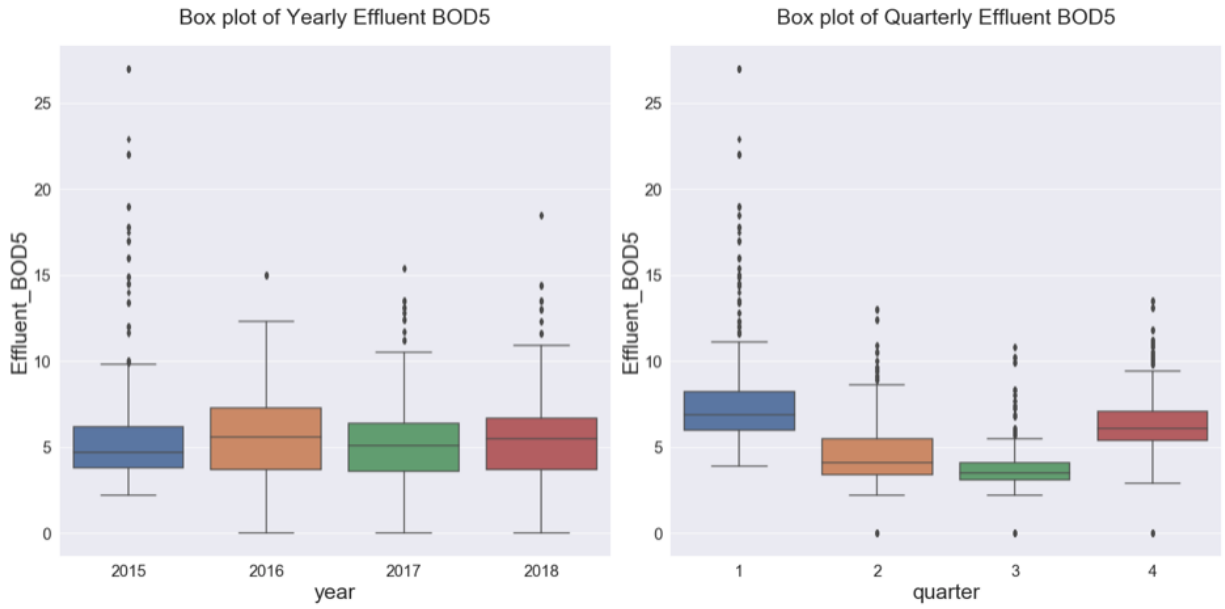


Figure 3.21 Box plot of effluent BOD_5 by year and quarter from 2015-2018.

In Figure 3.22, the mean effluent BOD_5 over days, weeks, months, quarters, and years are steady. Figure 3.23 displays the average means of effluent BOD_5 over the years, quarters, months, and days. The data points are closer to each other.

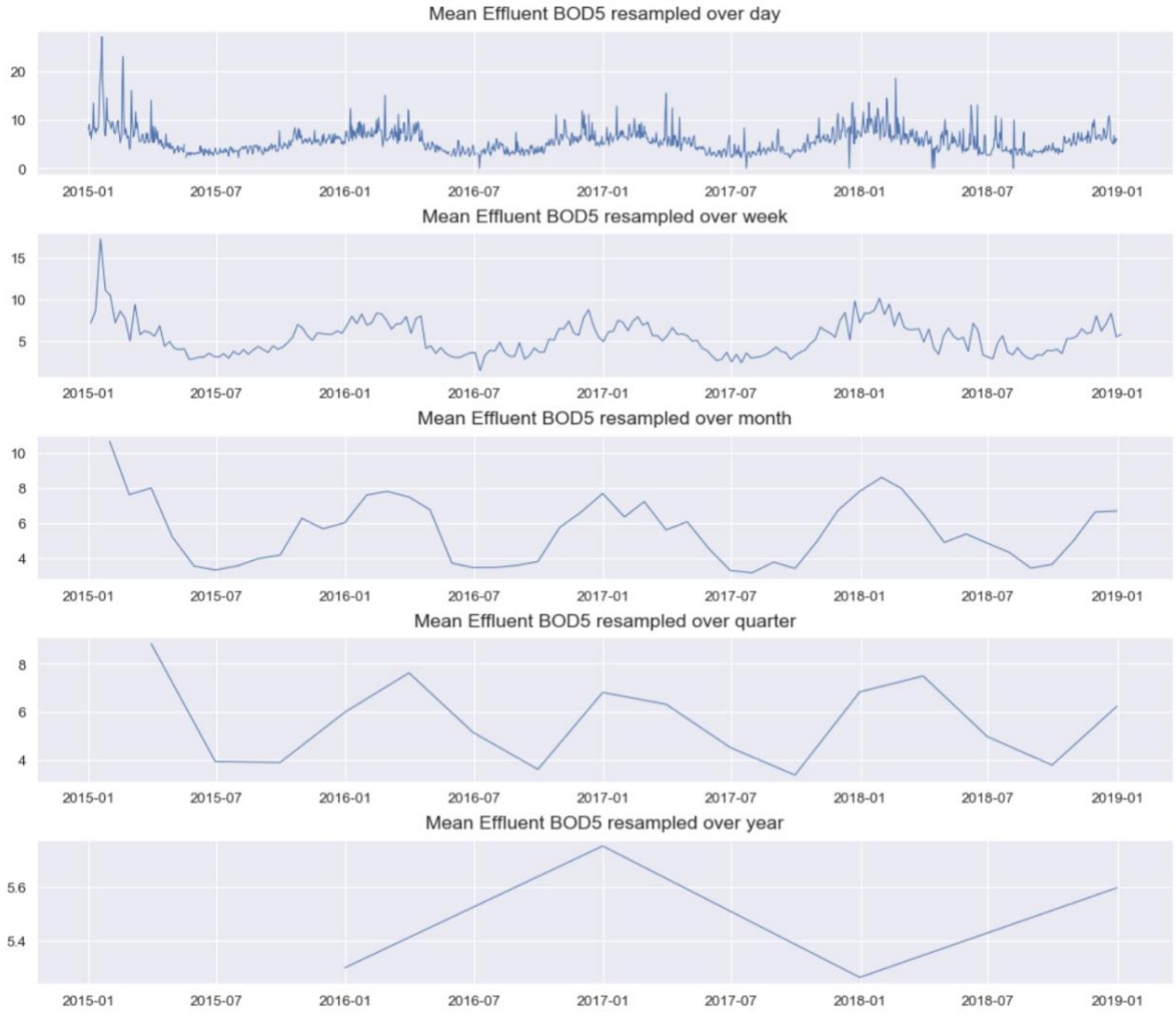


Figure 3.22 The mean effluent BOD_5 over a day, week, month, quarter, and year.

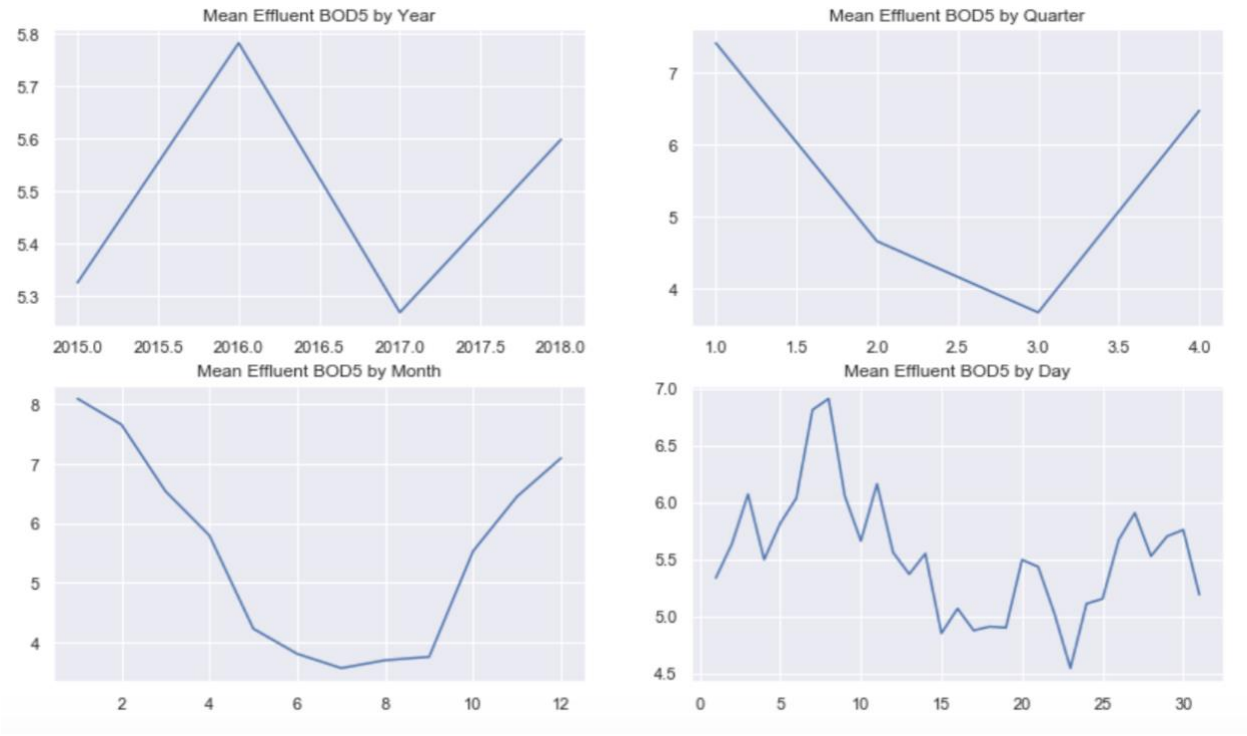


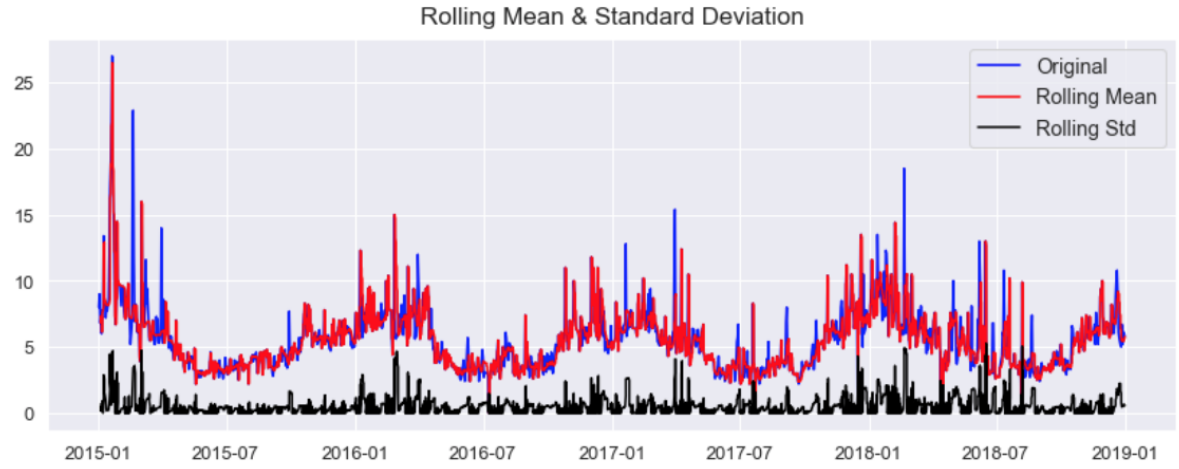
Figure 3.23 The average means of effluent BOD_5 over years, quarters, months, and days.

Finally, the Dickey–Fuller test with hypothesis testing was performed to determine if the data is stationary. When the data is stationary, it will make a model easier and faster to learn and better predict. The null hypothesis is that a unit root is present in an autoregressive model. The alternative hypothesis is usually stationarity or trend-stationarity. Stationary series has a constant mean and variance over time. The rolling average and rolling standard deviation of time series do not change over time.

- **Null Hypothesis (H0)**: It suggests the time series has a unit root, meaning it is non-stationary. It has some time-dependent structure; and
- **Alternate Hypothesis (H1)**: It suggests the time series does not have a unit root, meaning it is stationary. It does not have a time-dependent structure.

- P-value > 0.05: Accept the null hypothesis (H_0); the data has a unit root and is non-stationary; and
- P-value ≤ 0.05: Reject the null hypothesis (H_0); the data does not have a unit root and is stationary.

Figure 3.24 shows that P-value is less than 0.05. After running the data, the test statistics value was -12.1262. The more the negative test statistics means, the more the null hypothesis is rejected. Therefore, the data reject the null hypothesis H_0 with a significance level of less than 1%, which means the data is a stationary dataset and does not have a unit root.




```
<Results of Dickey-Fuller Test>
Test Statistic                  -12.1262
p-value                           0.0000
#Lags Used                       12.0000
Number of Observations Used   93812.0000
Critical Value (1%)              -3.4304
Critical Value (5%)              -2.8616
Critical Value (10%)             -2.5668
dtype: float64
```

Figure 3.24 The Dickey–Fuller test with hypothesis testing.

3.5 Conclusions and Recommendations

Big data analytics was performed to analyze data collected in the Nine Springs WWTP from 1996 to 2019. The methods include data collection, data understanding, data preparation, data mining, and evaluation. The following conclusions can be drawn:

- The background, goals, and unit processes of a WWTP must be studied to understand the dataset clearly.
- In data preparation, the first step was to select the data collection locations. The selected datasets should be the influent and effluent data. The second step is to clean datasets by filling or deleting missing values. In the Nine Springs WWTP, significant parameters were TSS, TP, TKN, NH_3N, and BOD_5, which are the essential parameters affecting wastewater treatment quality. Data visualization helped assess the relationship between influent and effluent.
- In the data preprocessing or data mining stage, the descriptive statistics function was applied to measure the average and distribution of the numerical values in the dataset. The correlation coefficient helped calculate the relationship among parameters. Effluent BOD_5 closely correlated to other parameters. The correlation values show that TSS, NH_3N, and TKN highly correlate to effluent BOD_5, which are 0.443, 0.428, and 0.342, respectively. TP has less correlation, which is 0.095. However, TP is one of the critical regulatory parameters, so it recommends applying as an input for model development.
- The normality test showed that effluent BOD_5 data rejected the null hypothesis, which means that the data is not a Gaussian distribution nor a normal distribution. In addition, Kurtosis and skewness testing can help determine normal distribution. The result showed

that Kurtosis is 5.205, implying the heaviness of the tails of the distribution, and skewness is 0.779, meaning the data are moderately skewed.

- Visualizing data using a box plot and graphical representation showed that the median effluent BOD_5 in 2014 is higher than the other years, which is not normal. Therefore, the data in 2014 should be removed and data in 2004 and 2019 because of the missing data.
- Finally, the Dickey-Fuller test was performed in the evaluation step to assess the data from 2015 to 2018. The result showed that the data rejected the null hypothesis H_0, implying that the data is stationary. Therefore, when the data has a clear trend and seasonality, it will fit a predictive model in the next chapter.

In conclusion, data analytics with statistical analysis is essential for analyzing and interpreting data, especially big data. This method will help find insights, remove unnecessary information, obtain a suitable dataset, and develop a precise predictive model.

4. RECURRENT NEURAL NETWORKS (RNN) FOR MODEL PREDICTION

4.1 Abstract

Artificial Intelligence (AI) has recently become one of the most powerful tools for applying in many fields. AI models human intelligence systems using machines or computer systems. Various AI applications involve expert systems, image recognition, speech identification, machine vision, and natural language processing (NLP). Artificial Neural Networks (ANNs) or Deep Neural Networks (DNNs) are the pieces of a computing system created to imitate the processes that the human brain analyzes and manages data. ANNs are the foundations of AI and can be applied to solve problems that are difficult for humans or even statistical analytics (Frankenfield, 2020)

DNN is one of the most powerful machine learning models capable of achieving outstanding performance on challenging problems. DNNs can be trained with supervised backpropagation when the labeled training set has enough information to specify the network's parameters, leading to excellent results (Sutskever et al., 2014). However, sequence data is a challenge for traditional DNNs because they require that the dimensionality of the inputs and outputs is known and fixed. Recurrent Neural Networks (RNNs) is a deep learning model designed explicitly for sequential data to introduce a feedback mechanism. However, simple RNNs can only pick up short-term or long-term architecture (Lamons et al., 2018). With a complex time series problem, both types of features (short-term or long-term) are needed. The solution is using Long Short-Term Memory (LSTM) cell. The LSTM architecture achieves better performance because of a gate in the cell that can control the process of memory, which is called the 'forget gate.'

Data from wastewater treatment operation is a sequence of historical data, which varies due to the environmental conditions. Therefore, this study implemented the simple RNNs and LSTM architecture to develop prediction models and evaluate the model performance to determine the most suitable model for sequence data in WWTPs.

4.2 Introduction

4.2.1 Background

Artificial Intelligence (AI) is the main concept of machine learning, deep learning, and ANNs. AI has become one of the most powerful tools to overcome difficult problems in various fields and solve complex real-world applications. AI is a branch of computer science dealing with the simulation of intelligent behavior in computers. It also means a computer system can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages (Nicholson, 2020).

Deep learning or DNNs are a collection of algorithms that have programmed new records in precision for various problems, e.g., sound recognition, image recognition, stock market prediction, recommender systems, and several other systems. The neural system of humans stimulates the fundamental concept of ANN (Haider et al., 2019). Multiple layers in neural networks are also called hidden layers, allowing the neural networks to learn from a large scale of the data. Recombining from one layer to more complex layers and passing the input data through these layers allow the computer to train data and result in higher accuracy.

Wastewater is accounting for many types of contaminators released into natural resources and the environment. Inadequate operation of a WWTP may bring increasing concern about the environment and public health problems. Water quality sensors and control systems in WWTPs are essential to maintain operational performance and keep the system stable. As a result, a large

amount of data is generated but, unfortunately, not systematically exploited. Meaningful information is the most critical factor in creating a predictive model in many areas. Still, the complexity of a WWTP results in uncertainty and variation of big data in the wastewater treatment systems. As many challenges increased, ANNs have been extensively studied as a reliable model for effective monitoring, predicting performance, and controlling the system and variables of the process in the complicated nonlinear and multivariable processes (Khademikia et al., 2016). Therefore, the ANN technique is essential to avoid process failure in WWTPs. Finally, ANNs have been developed to predict operational performance with a superior degree of accuracy and handle complex problems in wastewater treatment systems.

4.2.2 Ideal Predictive Model for a WWTP

The RNN is a natural generalization of feedforward neural networks to sequences. Given a sequence of inputs ($x_1, \ldots, x_T$), a standard RNN computes a series of outputs ($y_1, \ldots, y_T$) by iterating the following equation (Sutskever et al., 2014):

$$\begin{aligned} h_t &= \text{sigm}\left(W^{\text{hx}} x_t + W^{\text{hh}} h_{t-1}\right) \\ y_t &= W^{\text{yh}} h_t \end{aligned} \qquad 4.1$$

The RNN can easily map sequences to sequences whenever the alignment between the inputs the outputs are known ahead of time. However, it is unclear how to apply an RNN to problems whose input and output sequences have different lengths with complicated and non-monotonic relationships. In Figure 4.1 (Olah, 2015), an RNN, A, has an input x_t and output h_t. A loop allows information to persist and pass from one step of the network to the next, in which traditional neural networks cannot handle this.

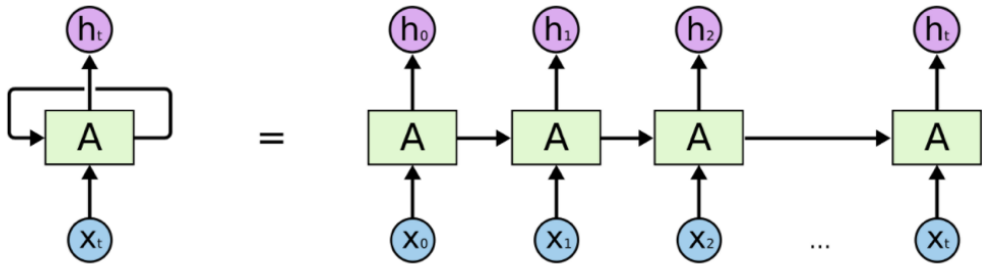


Figure 4.1 A Recurrent Neural Networks. (Source: Olah, 2015)

The most straightforward strategy for general sequence learning is to map the input sequence to a fixed-sized vector using one RNN and then map the vector to the target sequence with another RNN. While it could work in principle since the RNN is provided with all the relevant information, it would be challenging to train the RNNs due to long-term dependencies.

Long Short-Term Memory networks (LSTMs) are a particular type of RNN (Figure 4.2), capable of learning long-term dependencies (Olah, 2015). LSTM networks effectively perform many more tasks than the typical RNNs.

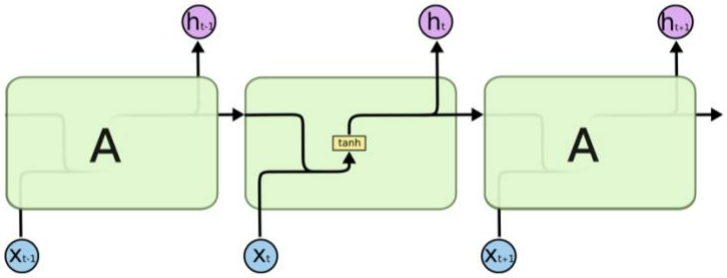


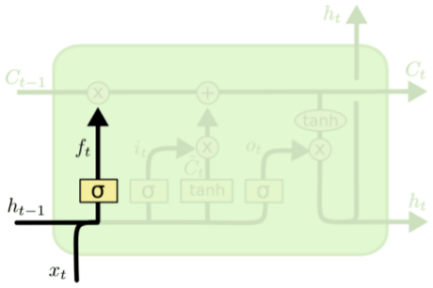
Figure 4.2 Long Short-Term Memory networks (LSTM). (Source: Olah, 2015)

LSTMs are known to learn problems with long-range temporal dependencies so that an LSTM network may succeed in this setting. The goal of the LSTM network is to estimate the conditional probability p ($y_1, \ldots, y_{T'}|x_1, \ldots, x_T$) where ($x_1, \ldots, x_T$) is an input sequence and $y_1, \ldots, y_{T'}$ is its corresponding output sequence whose length T′ may differ from T. The LSTM computes this

conditional probability by first obtaining the fixed dimensional representation v of the input sequence (x_1, . . ., x_T) given by the last hidden state of the LSTM, and then computing the probability of y_1, . . ., $y_{T'}$ with a standard LSTM-LM formulation whose initial hidden state is set to the representation v of x_1, . . ., x_T (Sutskever et al., 2014):

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1}) \quad 4.2$$

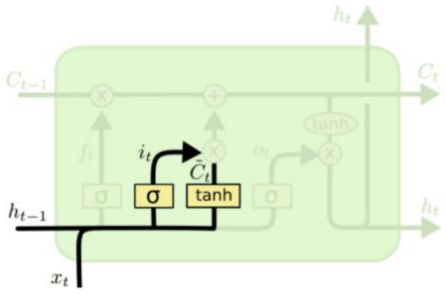
In LSTMs, the key to the memory cell is the gates, which include three gates – forget gate, input gate, and output gate. The first step in LSTM networks is the forget gate, which decides what data to discard from a cell. The decision is made by a sigmoid layer called the "forget gate layer." It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents "completely keep this," while zero means "completely get rid of this." Figure 4.3 (Olah, 2015) shows forget gate architecture and the equation.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 4.3 Forget gate architecture. (Source: Olah, 2015)

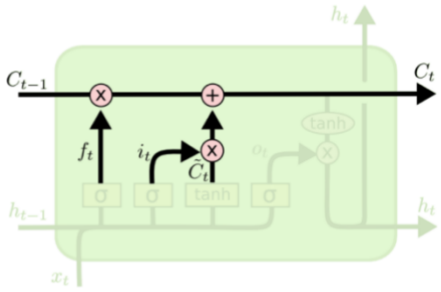
The next step is to decide the data that should be stored in the cell state. This consists of two parts. First, a sigmoid layer called the "input gate layer" decides which values should be updated. Second, a tanh layer creates a vector of new values, $\tilde{C}_t$, that could be added to the state. Lastly, the combination of these two steps creates an update to the state (Figure 4.4).



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 4.4 The cell state for the input gate. (Source: Olah, 2015)

To update the previous cell state, C_{t-1} is added into the new cell state, C_t, by multiplying the former state by f_t to forget the previous data that the model decides not to remember, and then, add $i_t * \tilde{C}_t$ (Note that * represents the element wise multiplication of the vectors). This formula is the new candidate values to update each cell state as shown in Figure 4.5 (Olah, 2015).



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 4.5 Update each cell state. (Source: Olah, 2015)

Finally, the output will be decided base on the cell state. First, a sigmoid layer will determine what parts of the cell state are going to be output. Then, the sigmoid gate puts the cell state through tanh (to push the values to be between -1 and 1) and multiplies it by the output so that the output will be decided as shown in Figure 4.6 (Olah, 2015).

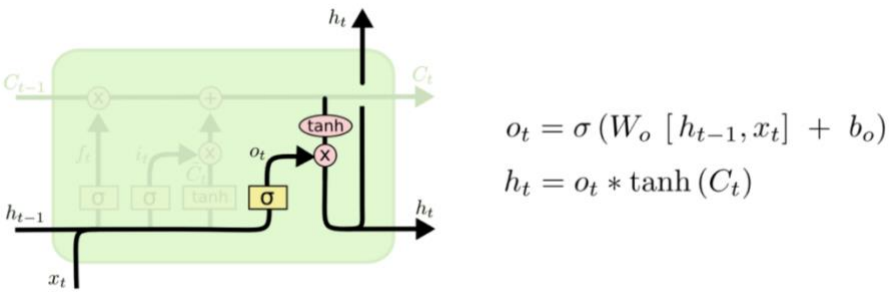


Figure 4.6 Output gate architecture. (Source: Olah, 2015)

4.2.3 Shortcomings of Previous Predictive Models

Monitoring and controlling wastewater treatment processes have increased consideration and led to developing several models for the biological treatment processes in WWTPs (e.g., ASM1, ASM2, ASM2d, and ASM3) (Henze et al., 2015). Still, the complex structures of these models involving large numbers of parameters that must be identified make them inappropriate for monitoring purposes (Harrou et al., 2018). For instance, the model ASM1 contained 13 nonlinear differential equations and 19 parameters, which are very difficult for computation (Dochain & Vanrolleghem, 2000).

Harrou et al. (2018) have studied a monitoring strategy using deep learning approaches, deep belief networks (DBNs), and a one-class support vector machine (OCSVM). However, when the data is highly noisy, false alarms might be generated during the fault detection task. As a result, deeper learning algorithms such as deep neural networks (DNNs), artificial neural networks (ANNs), and recurrent neural networks (RNNs) should be proposed to achieve complex input information in WWTPs.

Many researchers have recently developed various types of predictive models. Zhao et al. (2020) found that with the development of AI technology, the number of research publications of AI

application to wastewater treatment was 19 times greater in 2019 than in 1995, and papers had 36 more citations on average. However, most AI applications in WWTP are limited to ANN models. RNN models have not so far been applied to WWTP operations. Table 4.1 shows the use of AI models for operation management in WWTPs.

ANNs have been widely applied in operational wastewater treatment systems. However, there is a limited study on RNNs and LSTMs application in WWTPs.

4.2.4 Study Objective

The objective of the study is to develop RNN models to provide an accurate prediction of a WWTP. The models are established by applying simple RNN and RNN-LSTMs. The two types of models will be evaluated. In addition, the size of data, the number of epoch, and batch size are also compared to determine the most effective model for predicting wastewater treatment operation.

Table 4.1 Applications of AI models for operation management in WWTPs.

Treatment Process	AI Model	Model Performance (RMSE)	Reference
Anaerobic digestion	ANN-GA ANN	196.1 447.7	(Huang et al., 2016)
Aeration diffusion	ARMA-VAR	113.56	(Nadiri et al., 2018)
Aeration diffusion	BP-ANN GA-BP-ANN	303.51 232.6	(Man et al., 2019)
Anaerobic oxic biological	SDAE	5.94 1.27 1.26	(Shi & Xu, 2018)
Activated Sludge	PFA	0.25	(Yu et al., 2019)
Activated Sludge	SVM BP-ANN ANFIS RBF-ANN	1435.4 1445.9 1515.6 1501	(Najafzadeh & Zeinolabedini, 2019) (Zhao et al., 2020)

4.3 Materials and Methods

4.3.1 Data Preparation

The digital dataset was obtained from the Nine Springs WWTP in Madison Metropolitan Sewerage District (MMSD), which averagely treats 41 million gallons per day (mgd) of wastewater. Figure 4.7 represents the overview of an RNN modeling process.

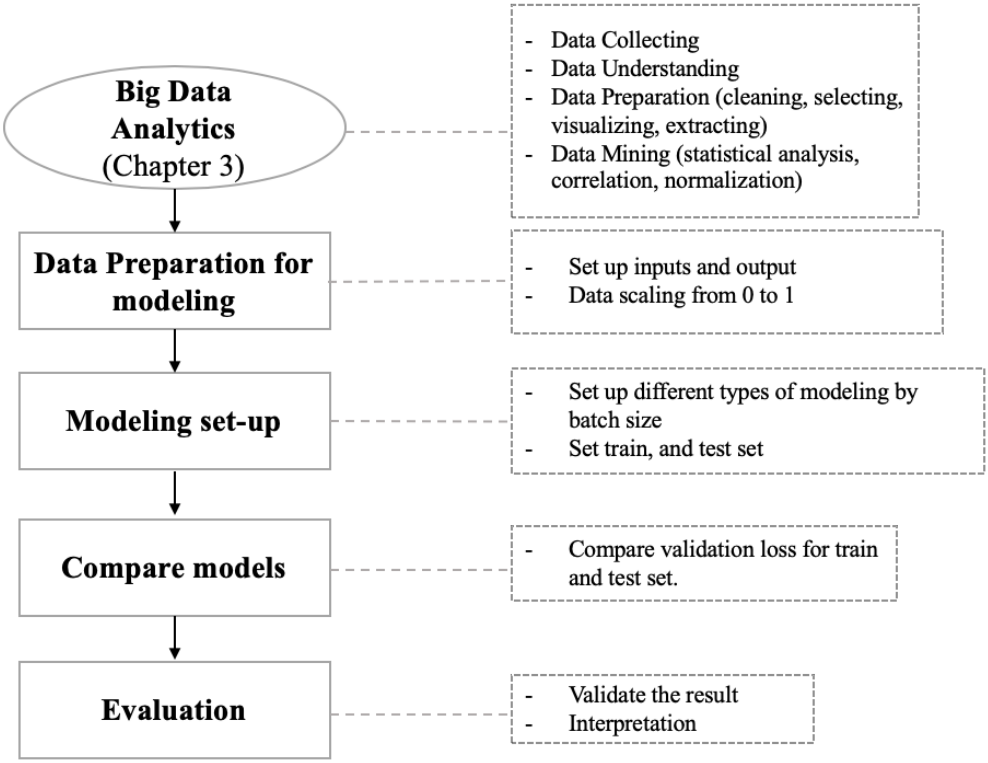


Figure 4.7 The overview of an RNN modeling process.

In the WWTP, biochemical oxygen demand (BOD_5) is one of the most important parameters used to evaluate the performance and meet the regulatory requirement. Therefore, five important influent quality parameters, BOD5, TSS, TP, TKN, and NH3-N, were selected as input variables to develop the model architecture. In addition, the flow rate and SVI were included as inputs for the daily prediction model. The outputs are the effluent of BOD_5 and other effluent parameters.

In the first part, two scenarios were used: data from 1996 to 2019 for validation and data from 2015 to 2018 for prediction of the model according to the findings from Chapter 3. Only BOD_5 data was applied in this step.

Daily wastewater treatment data were also calculated and collected to develop daily prediction models. Furthermore, the different number of epochs and batch sizes were assessed to determine

the most suitable RNN model. Figure 4.8 shows the architecture of the RNN model, which contains five important influent parameters and outputs.

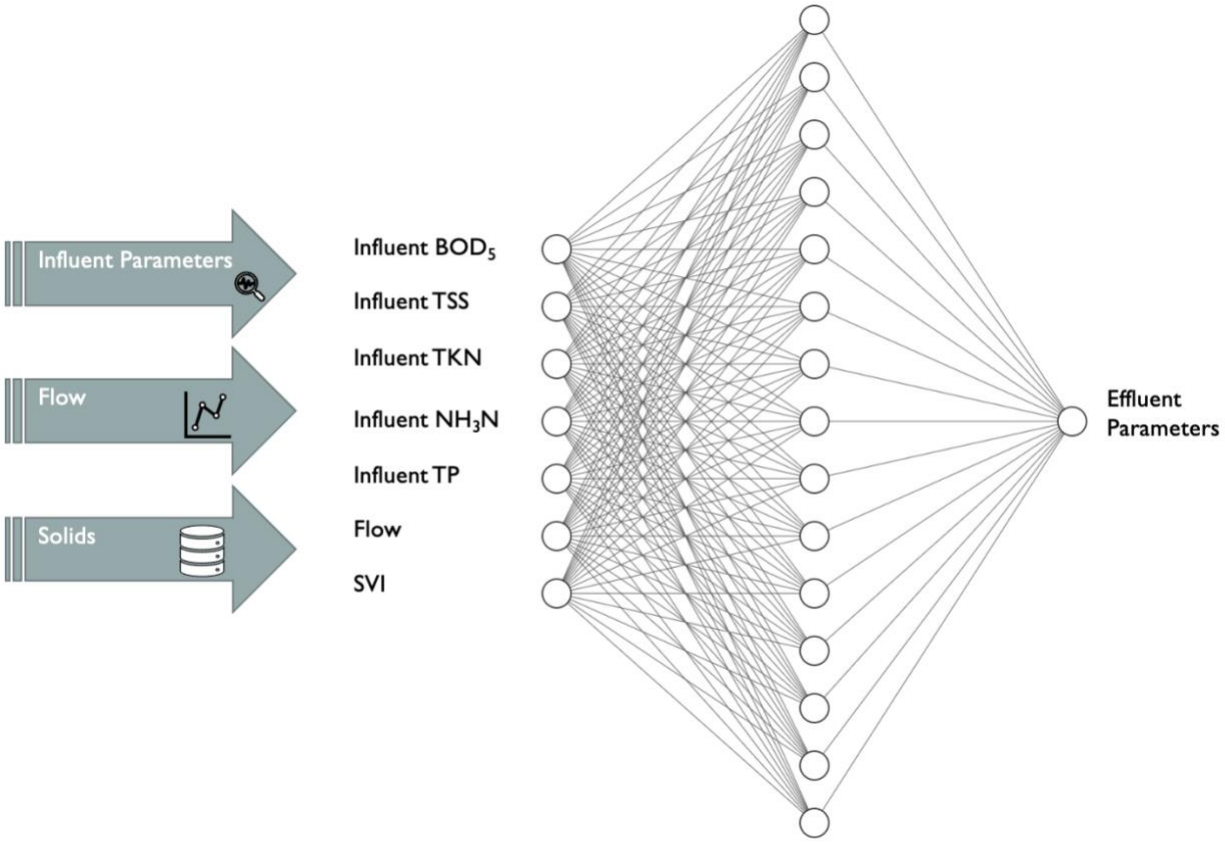


Figure 4.8 The architecture of the RNN model.

Data scaling

Data in prediction problems needs to be scaled when processing a neural network. When a network trains unscaled data with a wide range of values, it will slow down a learning rate and even prevent the network from effectively learning the data.

Normalization is a rescaling of the data within the range of 0 and 1 from the original range. In scaling, data is transformed using MinMaxScaler in the scikit-learn module. The equation below is rescaling (or min-max normalization), which is the simplest way to rescale data within the range [0, 1] or [−1, 1] (Han et al., 2011). The general formula for a min-max of [0, 1] is given as:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad 4.3$$

where x is an original value and x' is the normalized value.

Data scaling or data normalization is essential, especially when the data has input values with differing scales. The maximum value after applying the above formula is 1, and the minimum value is 0. Therefore, all the values will be between 0 and 1. Table 4.2 presents the data after rescaling.

Table 4.2 The result from data scaling.

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
reframed = series_to_supervised(scaled, 1, 1)
reframed.drop(reframed.columns[[7,8,9,10,11]], axis=1, inplace=True)
```

```
print(reframed.head())
```

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	var1(t)
1	0.296296	0.563376	0.173804	0.517385	0.783042	0.456763	0.333333
2	0.333333	0.423265	0.154176	0.454798	0.783042	0.439024	0.251852
3	0.251852	0.364614	0.165774	0.450626	0.783042	0.331486	0.251852
4	0.251852	0.364614	0.165774	0.349096	0.783042	0.331486	0.251852
5	0.251852	0.364614	0.165774	0.479833	0.783042	0.331486	0.251852

Root Mean Square Error (RMSE)

Root mean square error has been used as a standard statistical metric to measure model performance in meteorology, air quality, climate studies, and other research areas (Chai & Draxler, 2014). RMSE is the standard deviation of the prediction errors, which is the most popular measure of estimation accuracy to compare forecasting errors of different models, defined by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \quad 4.4$$

where n is a total number of samples in the model, and e is the model errors calculated as (e_i, i = 1, 2,... , n).

Mean Absolute Error (MAE)

Mean absolute error is a measure of errors between two observations. Y and X include comparisons of predicted versus observed**.** MAE has the same unit as the initial data, and it can be evaluated between models whose errors are computed in the same units (Hiregoudar, 2020). It is typically similar in magnitude to RMSE but slightly smaller. MAE is calculated as (Chai & Draxler, 2014):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

Coefficient of Determination (R^2)

R-squared (R^2) is a statistical measure representing the proportion of the variance for a dependent variable, which is explained by an independent variable in a regression model (Fernando, 2020). While correlation interprets the strength of the connection between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable (Hiregoudar, 2020). Thus, if the R^2 of a model is 0.50, then nearly half of the observed value can be explained by the model's inputs.

4.3.2 Development of Recurrent Neural Networks (RNNs) Models

An RNN model is suitable to make predictions on sequence data. The goal of this study is to develop a sequence prediction model that performs the best result. There are five steps to establish a simple RNN model (Figure 4.9) and LSTM model (Figure 4.10) as follows (Brownlee, 2017):

Figure 4.9 Five steps to develop a simple RNN model.

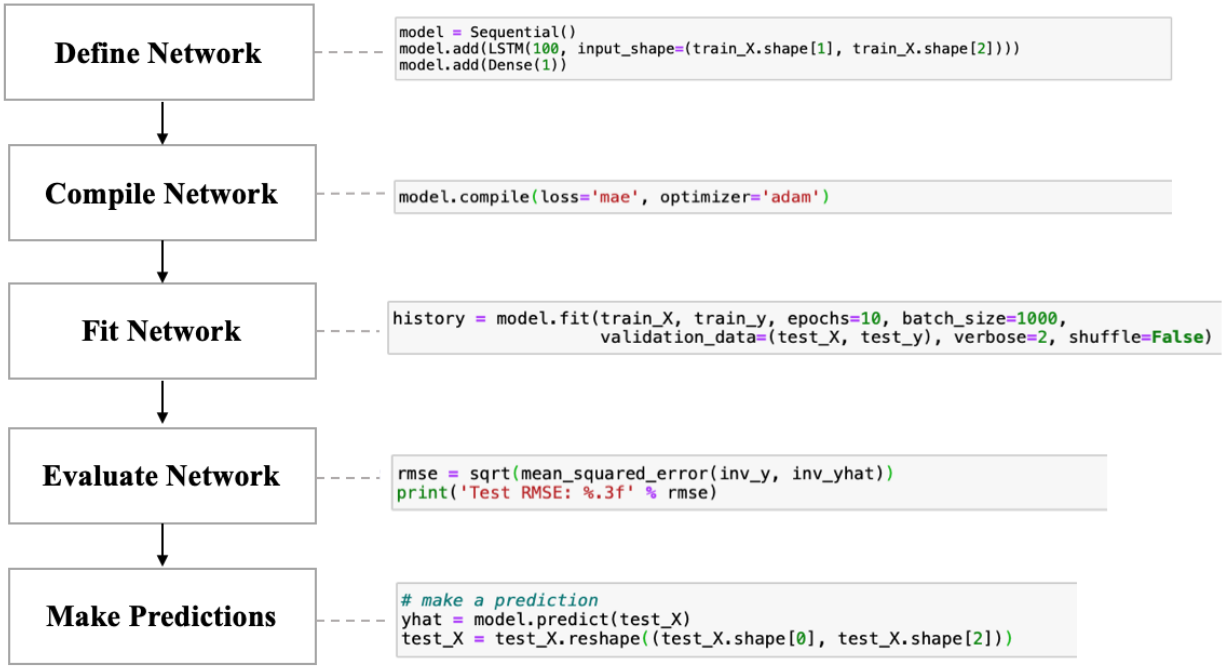
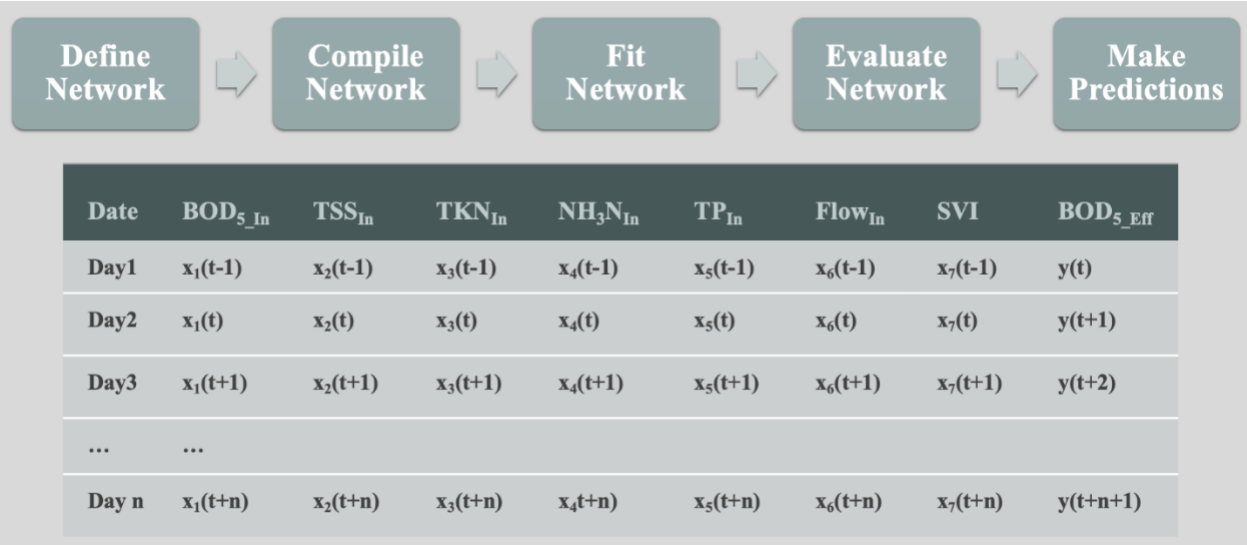


Figure 4.10 Five steps to develop an LSTM model.

- **Define Network**: Neural networks are a sequence of layers. The cover for these layers is the Sequential class. The steps are to obtain a sample of the sequential class, create layers, and add them together to be connected. A fully connected layer follows simple RNN or LSTM layers and adds Dense () for outputting a prediction. The first hidden layers must define the number of inputs, which must be three-dimensional: samples, time steps, and features.
- **Compile Network**: Complication expects to identify parameters to train a network. The optimization algorithm and loss function are required to train and evaluate the network, respectively. The standard optimization algorithms are stochastic gradient descent (sgd), adam, and RMSprop. The loss functions usually apply the mean squared error (mse) and mean absolute error (mae).
- **Fit Network**: After compiling the network, it needs to be fitted, which means adapting the weights on training data. Fitting the model expects the training dataset to specify a matrix and input patterns. The network exploits the Backpropagation for training and optimizing algorithms and loss function. The backpropagation requires a specific number of epochs, which means one passes through all data in the training set and updating the weights.
    - Epochs can be separated into groups of pattern pairs called a batch, which is a pass-through of a subset of samples in the training data after the network weights are updated. This step helps define the number of patterns in the network before the weights are updated within an epoch. It is an efficiency optimization, ensuring that not too many input data are loaded into a memory cell at a time. In addition, a verbose argument can be set to reduce the amount of information displayed when running the model.

- **Evaluate Network**: After training the network, then it needs to be evaluated. It is useful, especially for indicating the performance of a predictive model, because the network has seen all of the data before by separating data (no training). This will provide an estimate of the network's performance at making predictions for unseen data in the future.
- **Make Predictions**: After fitting and evaluating the model, the network will return the prediction values provided by the output layers of the network.

Figure 4.11 displays the steps of developing a time series model. The input parameters will be x variables to predict the output y. The model used 80% of the data for training and 20% for testing.



Date	BOD_{5_In}	TSS_{In}	TKN_{In}	NH_3N_{In}	TP_{In}	$Flow_{In}$	SVI	BOD_{5_Eff}
Day1	$x_1(t-1)$	$x_2(t-1)$	$x_3(t-1)$	$x_4(t-1)$	$x_5(t-1)$	$x_6(t-1)$	$x_7(t-1)$	$y(t)$
Day2	$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t)$	$x_5(t)$	$x_6(t)$	$x_7(t)$	$y(t+1)$
Day3	$x_1(t+1)$	$x_2(t+1)$	$x_3(t+1)$	$x_4(t+1)$	$x_5(t+1)$	$x_6(t+1)$	$x_7(t+1)$	$y(t+2)$
…	…							
Day n	$x_1(t+n)$	$x_2(t+n)$	$x_3(t+n)$	$x_4t+n)$	$x_5(t+n)$	$x_6(t+n)$	$x_7(t+n)$	$y(t+n+1)$

Figure 4.11 Steps to develop a time series model.

4.4 Results and Discussion

In this study, the historical data of the Nine Springs WWTP was used in developing the AI model. The values of parameters in wastewater treatment from 2015 to 2018 are summarized in Table 4.3.

Table 4.3 The values of water quality parameters in wastewater treatment from 2015 to 2018.

Parameters	Unit	Maximum	Minimum	Mean	SD
$BOD_{5,i}$	mg/L	400.00	93.10	246.53	39.44
$BOD_{5,e}$	mg/L	27.00	0.00	5.48	2.34
TSS_i	mg/L	1170.00	49.20	222.72	60.76
TSS_e	mg/L	22.30	0.00	4.74	1.67
TP_i	ppm	11.40	2.38	5.68	0.98
TP_e	ppm	1.12	0.05	0.31	0.12
TKN_i	ppm	90.00	18.10	44.80	7.56
TKN_e	ppm	9.65	0.23	1.79	0.49
NH_3N_i	ppm	40.10	0.00	28.26	4.01
NH_3N_e	ppm	7.22	0.00	0.26	0.36

ppm: Parts per Million

Performance Comparison

The first model performance comparison applied discrete big data from 1997 to 2019 and 2015 to 2018, separating into two scenarios. Scenario 1 has 3,703,522 samples of data, and scenario 2 has 82,767 samples. The models were performed by applying a simple RNN model and LSTM model with the different number of epochs and batch size. The input parameters are influent parameters, TSS, TP, TKN, and NH_3N. The output is the effluent BOD_5. The performance of models includes time and rooted mean squared errors (RMSE). Table 4.4 shows the effluent prediction models using simple RNN and LSTM models. Scenario 2, data from 2015 to 2018, achieves better timing and accuracy in both models when using a large number of epochs and small batch sizes. The

optimum was when specifying 20~50 epochs and 100 batch sizes with 1~2 seconds in each epoch and RMSE of 0.3~08.

Table 4.4 Performances of the effluent BOD_5 prediction models.

Model	Scenario	Training Epoch	Batch Size	Time (Seconds)	RMSE
Simple RNN model	Scenario 1: 3,703,522 samples (Data from 1997-2019)	10	1000	9-14	2.298
		20	100	37-84	4.688
		50	100	40-53	1.702
	Scenario 2: 82,767 samples (Data from 2015-2018)	10	1000	0-1	1.396
		20	100	1-3	0.390
		50	100	1-4	0.361
LSTM model	Scenario 1: 3,703,522 samples (Data from 1997-2019)	10	1000	10-15	1.888
		20	100	48-92	1.638
		50	100	49-62	1.529
	Scenario 2: 82,767 samples (Data from 2015-2018)	10	1000	0-2	1.888
		20	100	2-3	0.778
		50	100	1-2	0.452

Figure 4.12 represents the train and test loss plot over the epochs to evaluate the best two predictive models, simple RNN and LSTM, with a training epoch of 50 and batch size of 100. The result shows that the LSTM model fitted the data well.

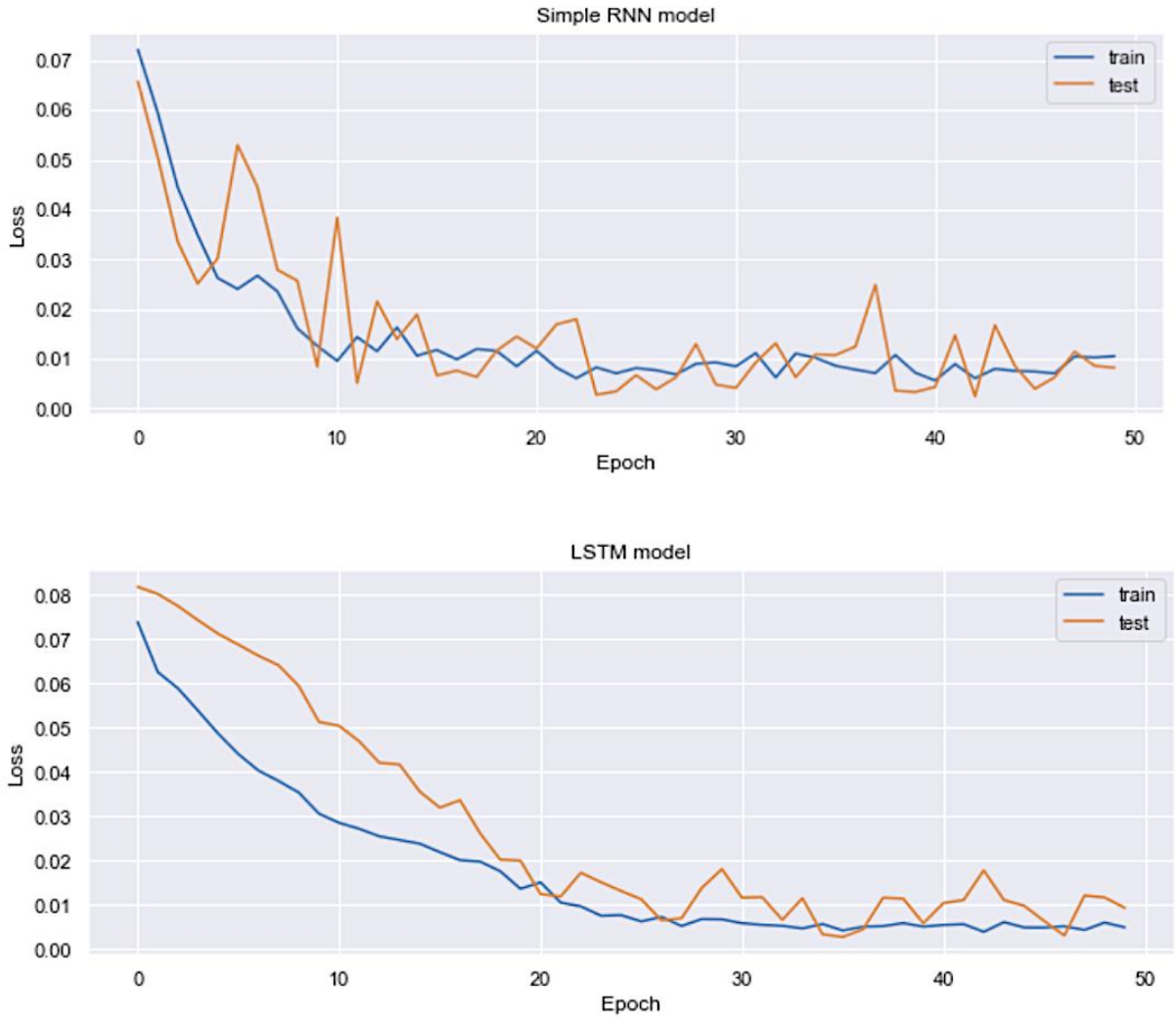


Figure 4.12 Train and test loss over epoch for effluent BOD_5 prediction models.

A good fit is determined by a training and validation loss that decreases to the point of constancy with a minimal gap between the two lines: training loss and validation loss (Brownlee, 2019). Moreover, the loss of the model should be lower on the training set than the test set. Thus, the LSTM model tends to be better than the RNN model because the final test loss is above the training

loss. However, the simple RNN model achieved a better RMSE of 0.361 than the LSTM model with an RMSE of 0.452 (Table 4.4).

Figure 4.13 shows the original dataset of effluent BOD_5 from January 1, 2015, to December 31, 2018. The data was collected hourly or daily. If the data was collected at the same interval, the prediction result would be more clearly seen.

Figure 4.14 shows the prediction result of effluent BOD_5 from 2015 to 2018 using the simple RNN algorithm. The original value is very close to the predicted value, which has an RMSE of 0.361. RMSE of 0.361 in the simple RNN model for big data from 2015 to 2018 is the smallest RMSE value. It means that the model is optimal and predicts an excellent in-sample fit.

Figure 4.15 is the result of effluent BOD5 from 2015 to 2018 from the LSTM model. The prediction result is very close to the original data. The graph shows that the prediction model is precise. The RMSE of the model is 0.452, which is also a minimal value. The low RMSE value presents that the model makes smaller errors, such as overfitting or underfitting. Therefore, the original dataset from 2015 to 2018 is a good fit dataset. The models can learn and remember the pattern so they can accurately predict the effluent BOD5. The next step is to develop a model for other important effluent parameters – TP, TKN, TSS, and NH3N. The model applied 100 batch sizes and 50 epochs, which create the best model performance. The summary of the accuracy of each model is shown below in Table 4.5. In addition, Figures 4.14 to 4.23 show the results of all effluent parameters models.

Table 4.5 The summary of the accuracy of effluent parameters models.

Figure	Model Architecture	Effluent Parameters	RMSE	MAE	R^2
4.14	Simple RNN	BOD_5	0.247	0.052	0.985
4.15	LSTM	BOD_5	0.246	0.058	0.985
4.16	Simple RNN	TP	0.019	0.004	0.981
4.17	LSTM	TP	0.018	0.003	0.981
4.18	Simple RNN	TKN	0.091	0.066	0.956
4.19	LSTM	TKN	0.067	0.027	0.976
4.20	Simple RNN	TSS	0.319	0.051	0.980
4.21	LSTM	TSS	0.320	0.074	0.980
4.22	Simple RNN	NH_3N	0.054	0.025	0.955
4.23	LSTM	NH_3N	0.056	0.029	0.952

According to the summary table above, Figure 4.16 is the prediction result of TP from the simple RNN model, and Figure 4.17 is the prediction result of TP from the LSTM model. Both models achieved the RMSE of ~0.02 and the MAE of 0.004, which are very low. It means that the models can predict with an error of less than 0.02. Other prediction models can also achieve very low errors. The prediction of TKN for both models achieved the RMSE of 0.09 and the MAE of 0.07. The RMSE and MAE for the prediction of TSS were 0.032 and 0.05, respectively. Lastly, the prediction of NH_3N had an RMSE of 0.06 and an MAE of 0.03. Thus, all the effluent prediction models achieved high accuracy with very low RMSE and MAE values. Moreover, R^2 score was close to 1, which means the model performance is very high.

The second model evaluation used daily data to develop a daily prediction model. In this model, five important influent quality parameters (BOD_5, TSS, TP, TKN, and NH_3-N), flow rate, and SVI

(Sludge Volume Index) were applied, and the daily data from 2015 to 2018 was used. The data has 1,444 rows of data. Table 4.6 shows the general descriptive statistics of the daily data from 2015 to 2018.

Table 4.6 General descriptive statistics of the daily data from 2015 to 2018

	count	mean	std	min	25%	50%	75%	max
Influent_BOD5	1444.00000	246.49903	38.58584	93.10000	222.00000	247.00000	271.00000	400.00000
Influent_TSS	1444.00000	218.67902	31.18781	114.00000	200.00000	218.00000	236.00000	366.00000
Influent_TKN	1444.00000	44.67396	5.70345	18.20000	41.30000	44.75000	48.20000	68.70000
Influent_NH3-N	1444.00000	28.38438	3.68645	0.00000	26.40000	28.60000	30.70000	40.10000
Influent_TP	1444.00000	5.71461	0.78715	2.38000	5.25750	5.78000	6.25000	9.34000
Effluent_BOD5	1444.00000	5.48303	2.46816	0.00000	3.70000	5.10000	6.60000	27.00000
Effluent_TSS	1444.00000	4.75886	1.67982	0.00000	3.90000	4.50000	5.20000	22.30000
Effluent_TKN	1444.00000	1.82546	0.62838	0.23000	1.52000	1.70000	1.95000	9.65000
Effluent_NH3-N	1444.00000	0.28913	0.51856	0.00000	0.09000	0.15000	0.29000	7.22000
Effluent_TP	1444.00000	0.31716	0.12200	0.05000	0.24000	0.29000	0.35000	1.12000
Flow_In (MGD)	1444.00000	41.44002	5.92495	32.15150	38.14920	40.23465	42.93342	112.94340
SVI_Plant1	1444.00000	95.46479	10.88372	59.32200	88.20712	94.60648	102.10311	139.78490

There are only about 1,400 rows of data, while original big data from 2015 to 2018 has about 82,000 rows of data. The same models, simple RNN and LSTM, were implemented. Figures 4.24 and 4.25 show the daily effluent BOD_5 prediction result using the Simple RNN and LSTM models, respectively. The result shows the prediction has less accuracy than the original big data model.

The RMSE of the simple RNN model for daily BOD_5 prediction is 1.827, and the MAE is 1.101, which are higher than the discrete big data prediction, RMSE of 0.247 and MAE of 0.052, respectively. The RMSE of the daily BOD_5 with the LSTM model is 1.842, and the MAE is 1.091, which are also higher than the first models using the discrete big data (RMSE of 0.247 and MAE of 0.058). Figures 4.26 to 4.35 show the daily prediction of other influent parameters, TP, TKN, TSS, NH_3N, and SVI, respectively.

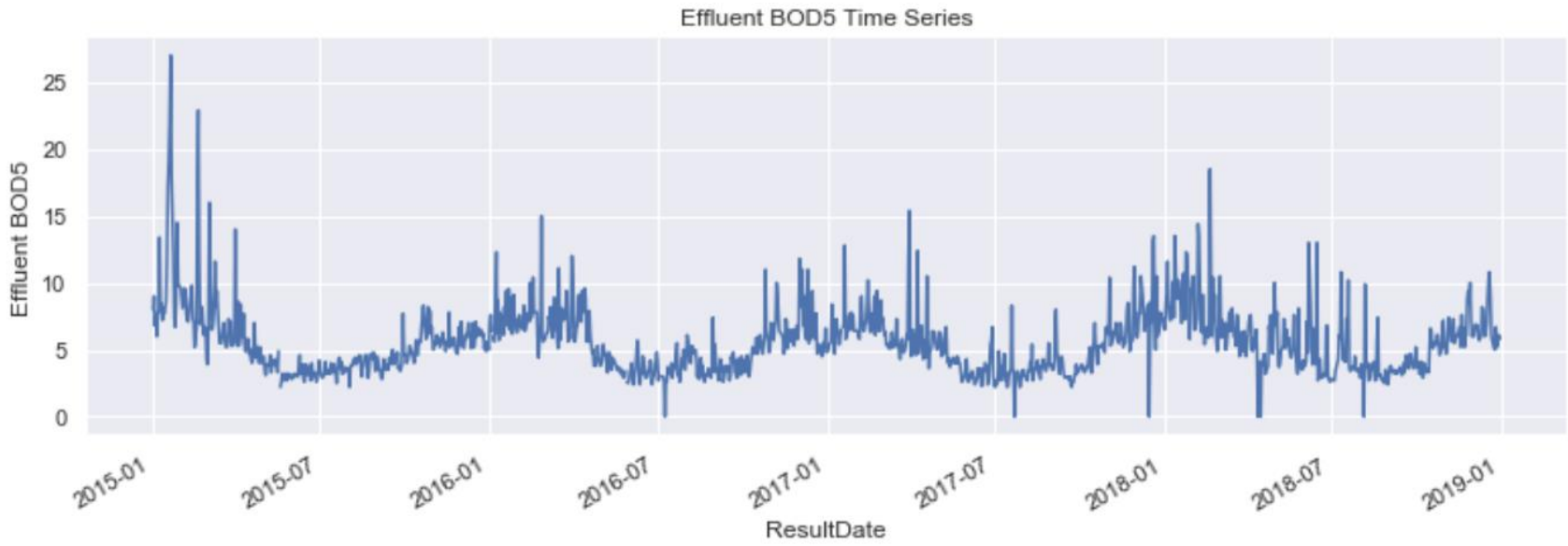


Figure 4.13 The original data of effluent BOD_5 from 2015 to 2018.

```
Test RMSE: 0.247
Test MAE: 0.052
Test r2 score: 0.985
```

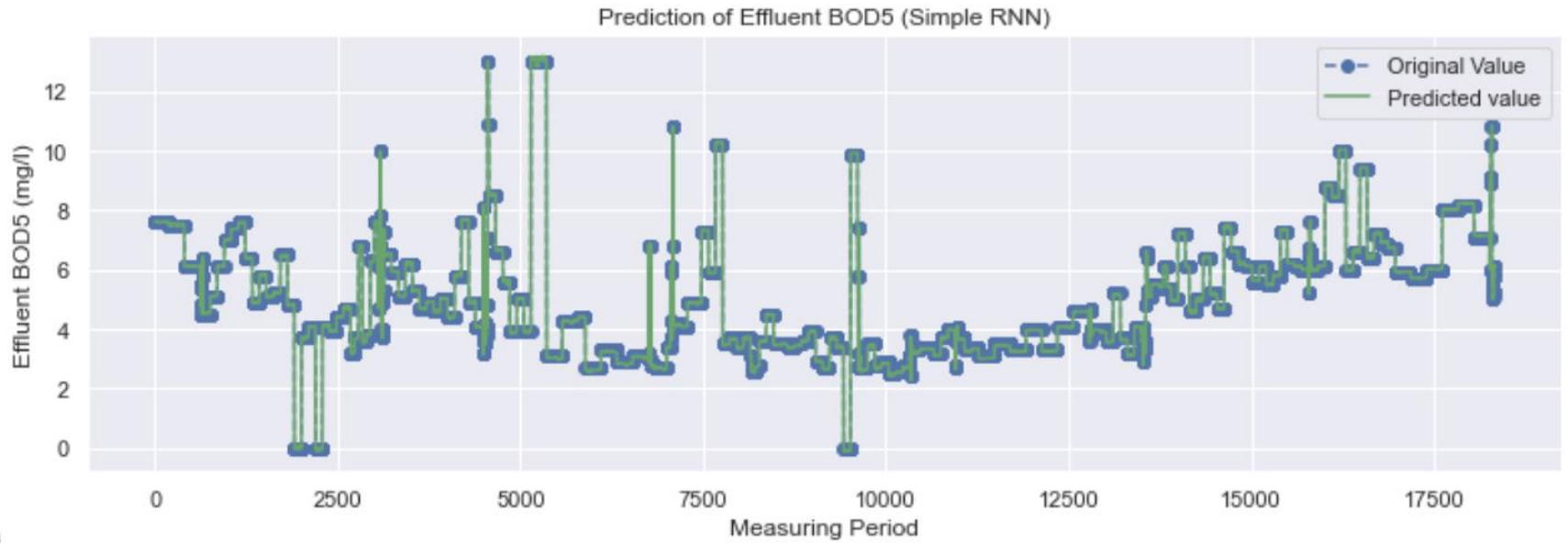


Figure 4.14 The prediction of effluent BOD_5 from 2015 to 2018 using the simple RNN model.

```
Test RMSE: 0.246
Test MAE: 0.058
Test r2 score: 0.985
```

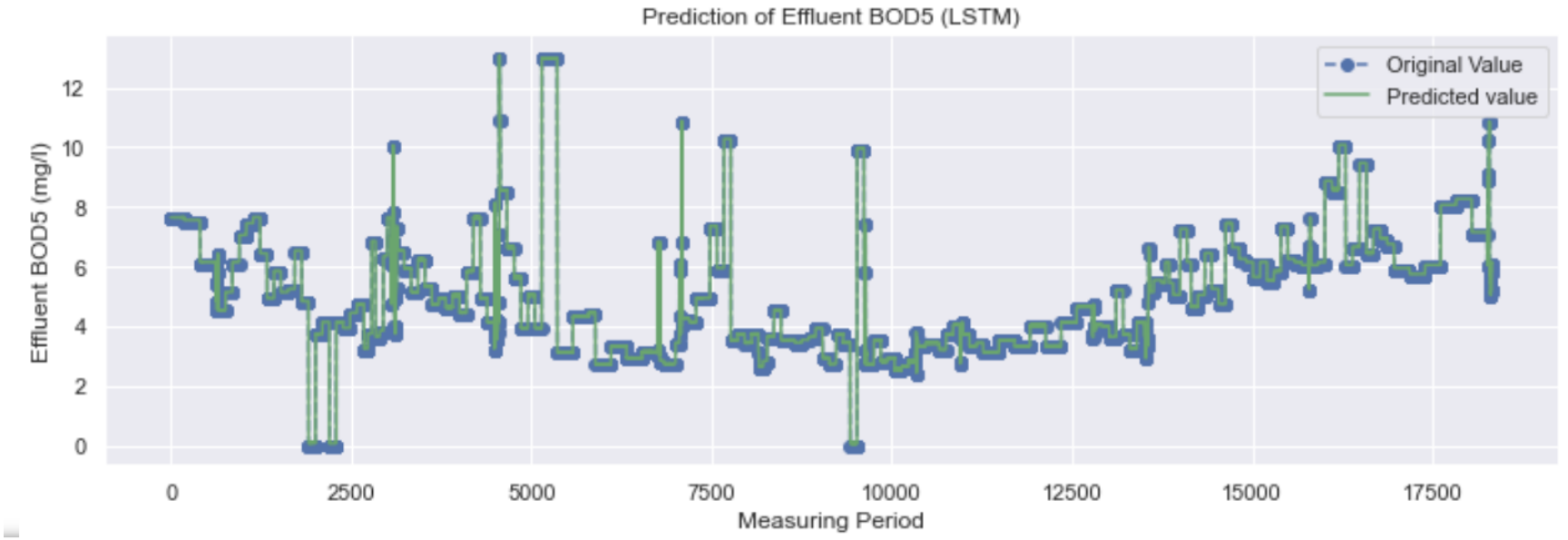


Figure 4.15 The prediction of effluent BOD_5 from 2015 to 2018 using the LSTM model.

```
Test RMSE: 0.019
Test MAE: 0.004
Test r2 score: 0.981
```

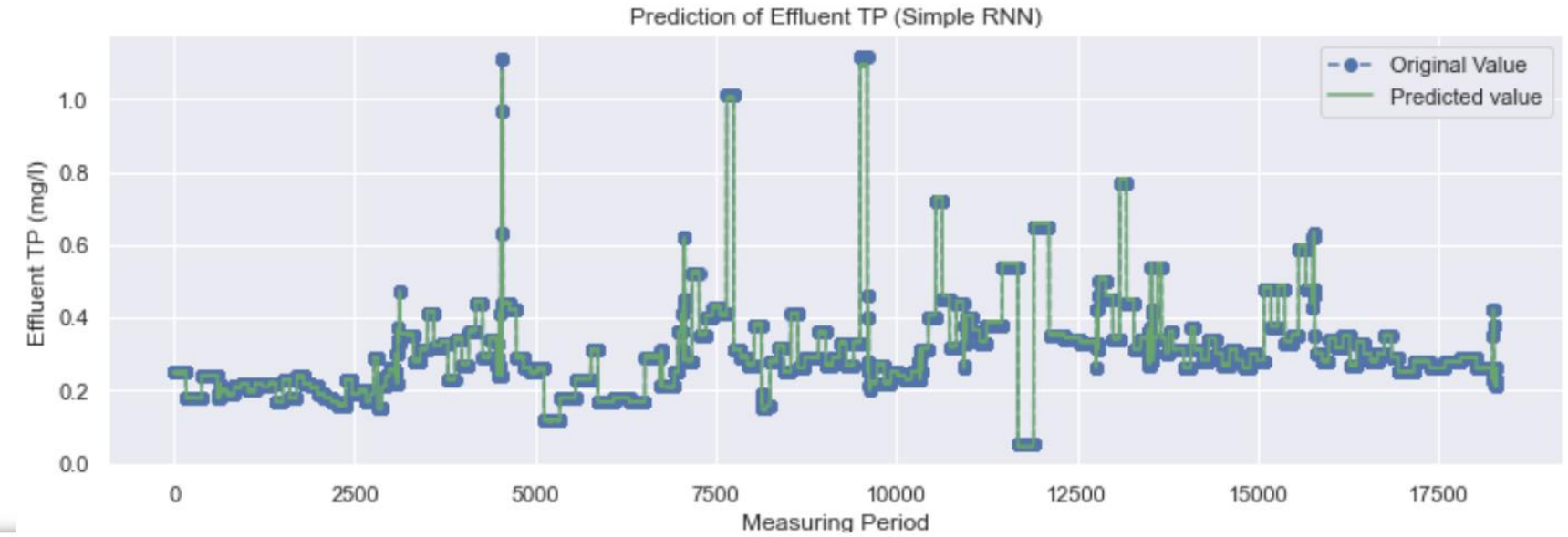


Figure 4.16 The prediction of effluent TP from 2015 to 2018 using the simple RNN model.

```
Test RMSE: 0.018
Test MAE: 0.003
Test r2 score: 0.981
```

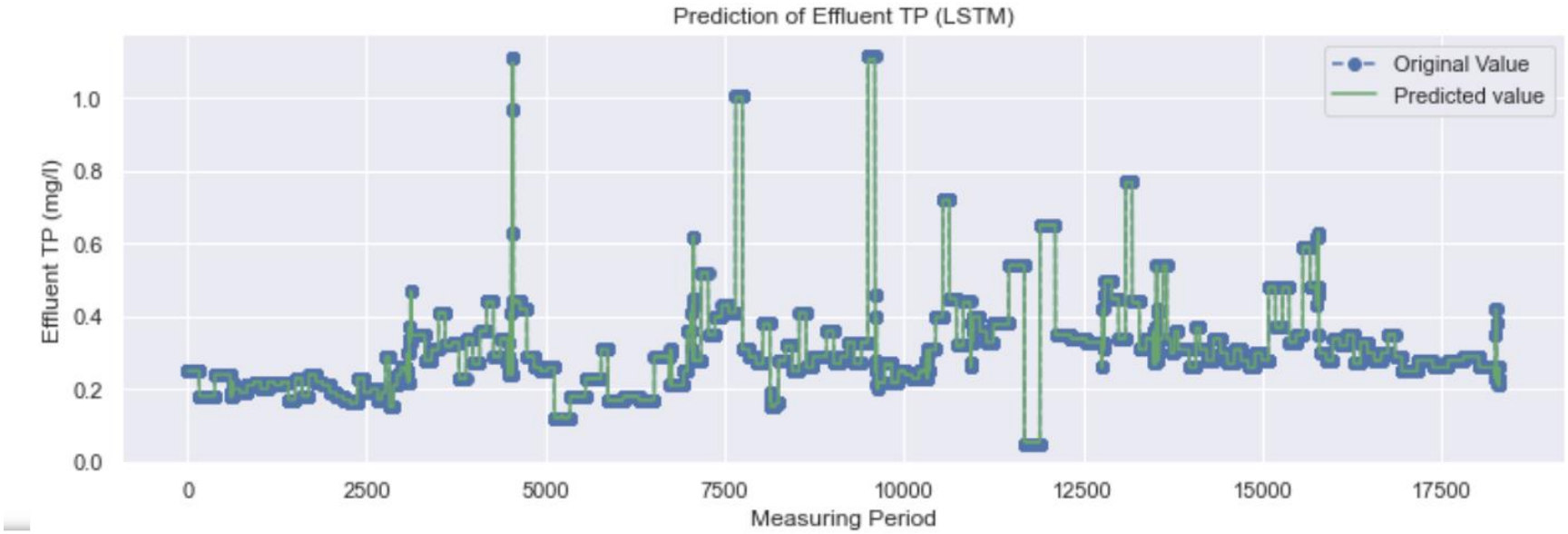


Figure 4.17 The prediction of effluent TP from 2015 to 2018 using the LSTM model.

```
Test RMSE: 0.091
Test MAE: 0.066
Test r2 score: 0.956
```

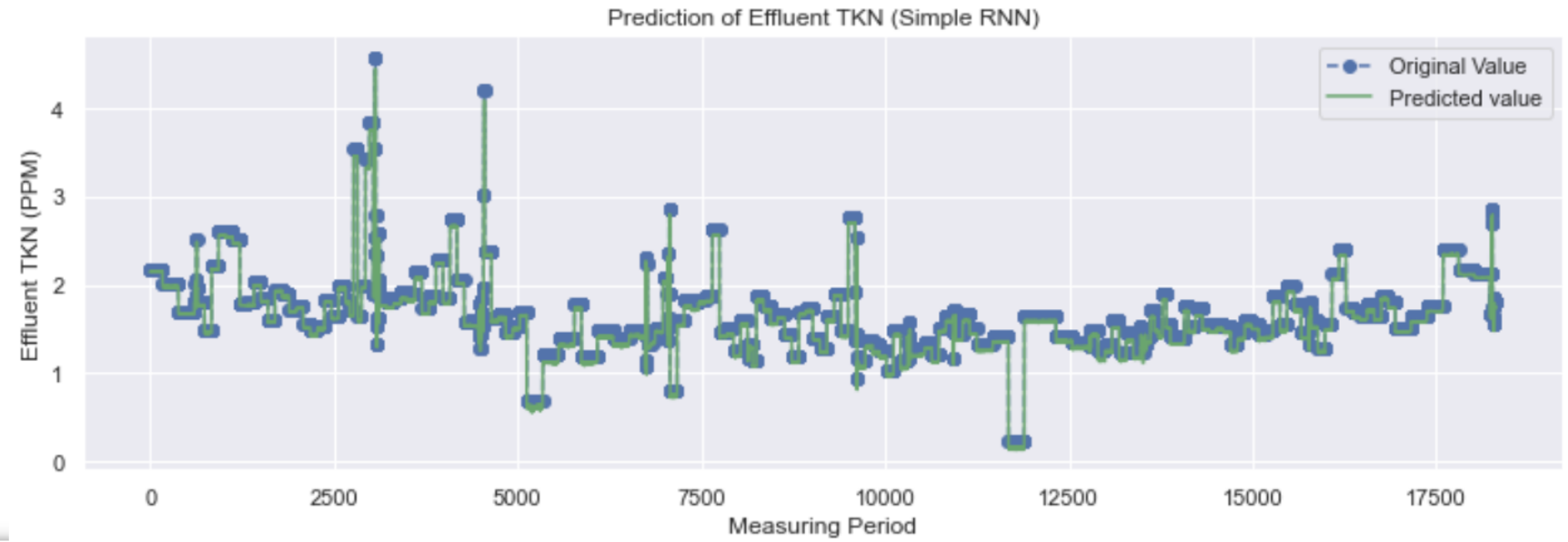


Figure 4.18 The prediction of effluent TKN from 2015 to 2018 using the simple RNN model.

```
Test RMSE: 0.067
Test MAE: 0.027
Test r2 score: 0.976
```

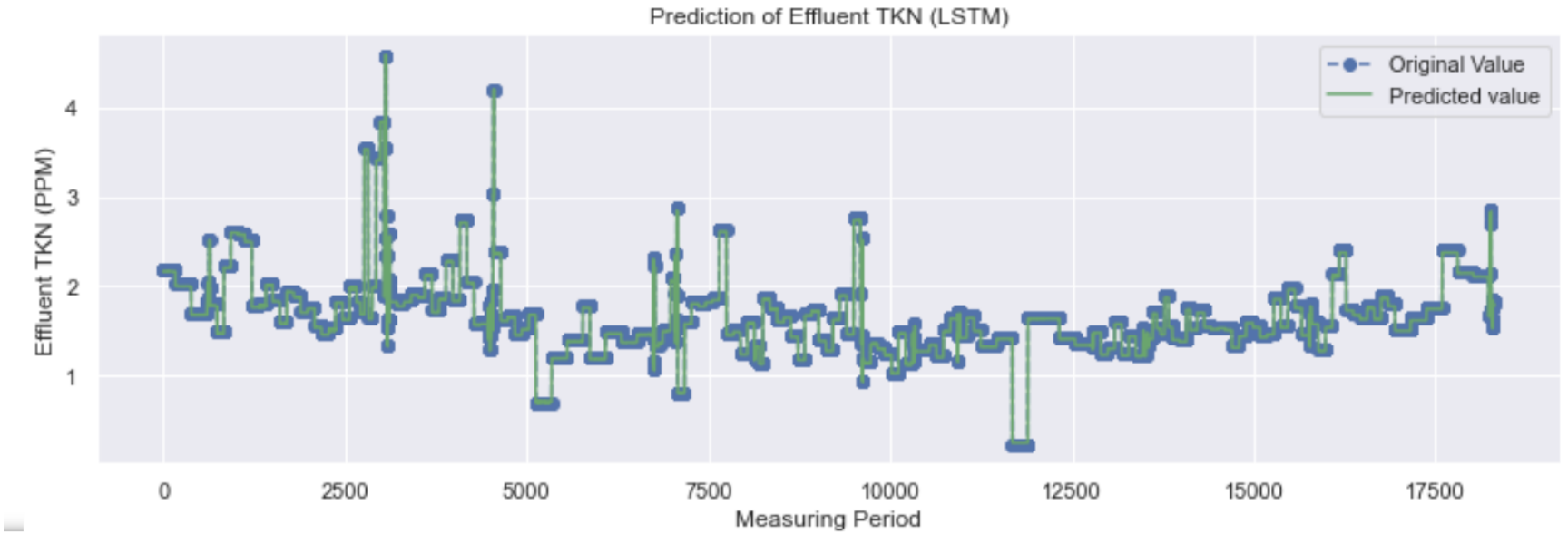


Figure 4.19 The prediction of effluent TKN from 2015 to 2018 using the LSTM model.

```
Test RMSE: 0.319
Test MAE: 0.051
Test r2 score: 0.980
```

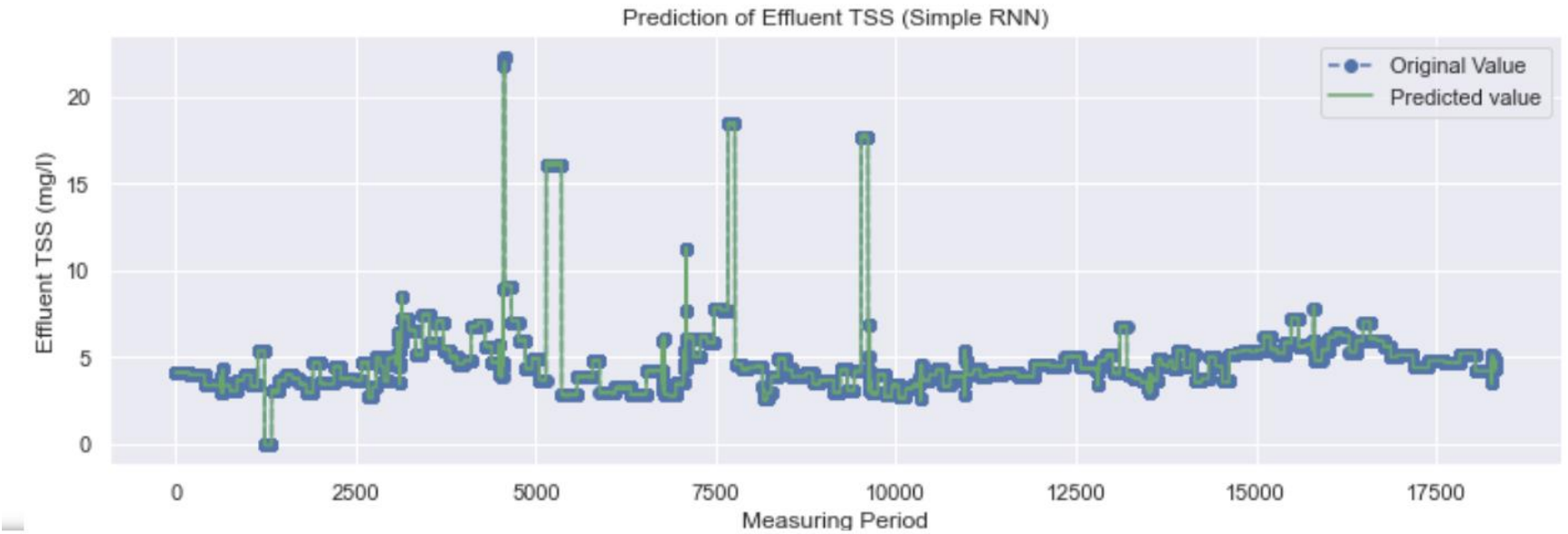


Figure 4.20 The prediction of effluent TSS from 2015 to 2018 using the simple RNN model.

```
Test RMSE: 0.320
Test MAE: 0.074
Test r2 score: 0.980
```

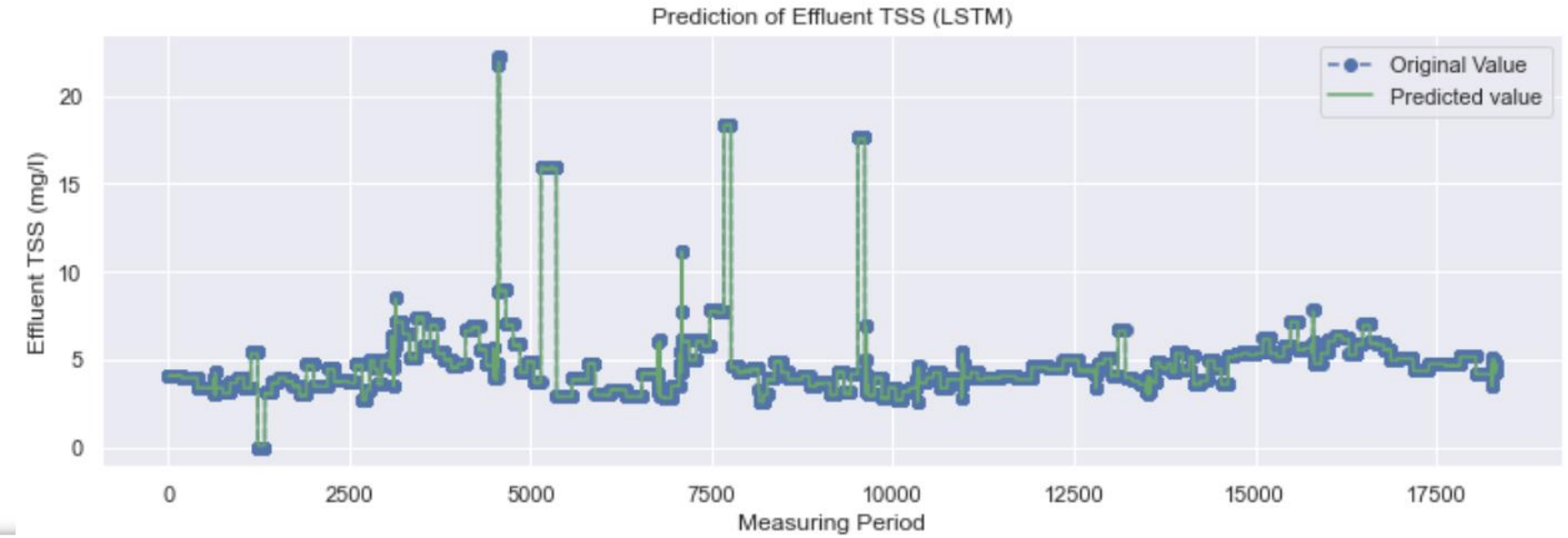


Figure 4.21 The prediction of effluent TSS from 2015 to 2018 using the LSTM model.

```
Test RMSE: 0.054
Test MAE: 0.025
Test r2 score: 0.955
```

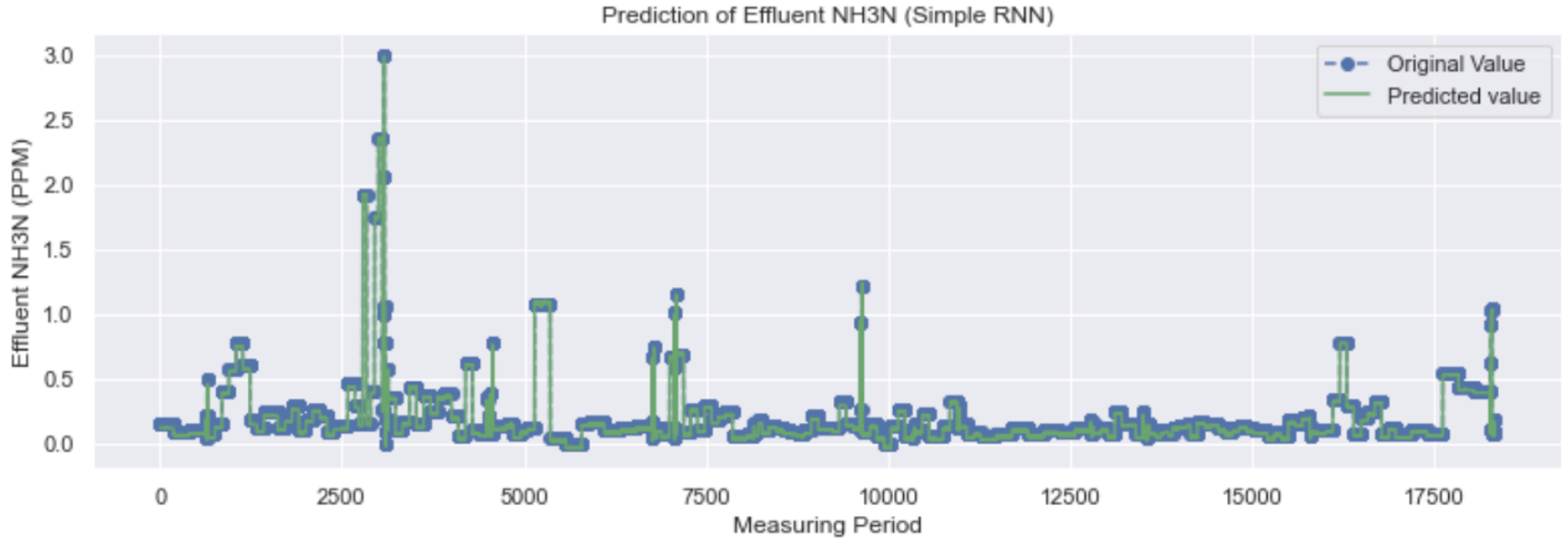


Figure 4.22 The prediction of effluent NH_3N from 2015 to 2018 using the simple RNN model.

```
Test RMSE: 0.056
Test MAE: 0.029
Test r2 score: 0.952
```

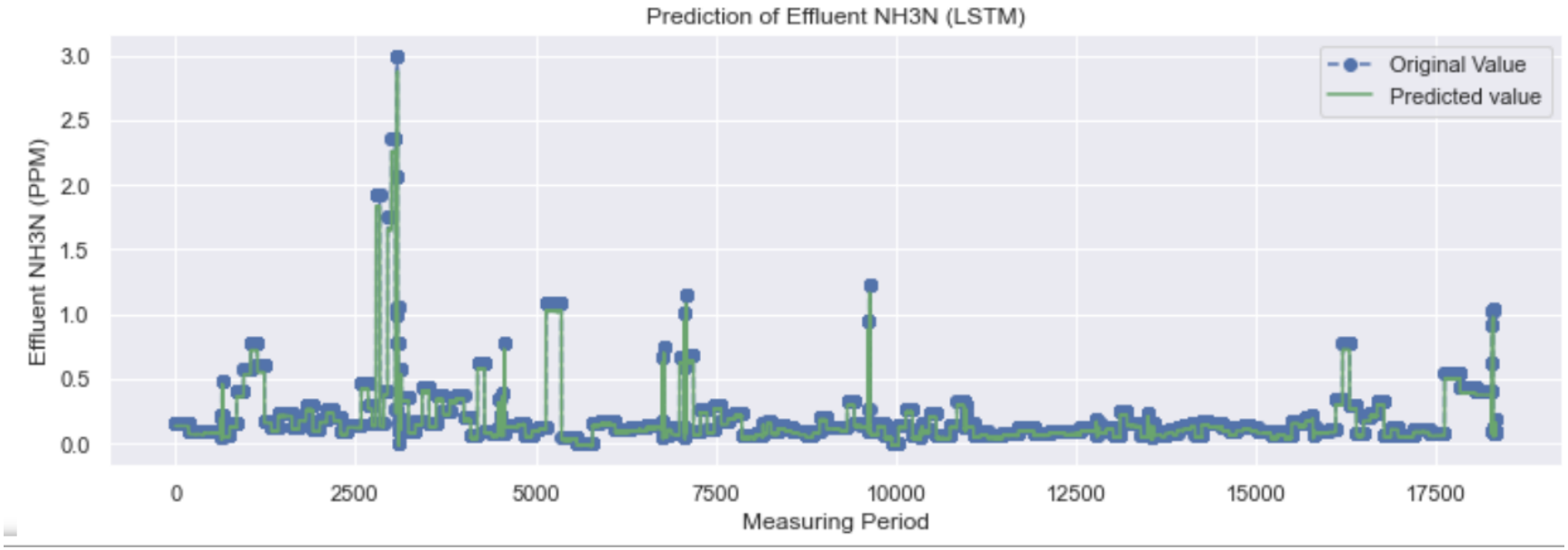


Figure 4.23 The prediction of effluent NH_3N from 2015 to 2018 using the LSTM model

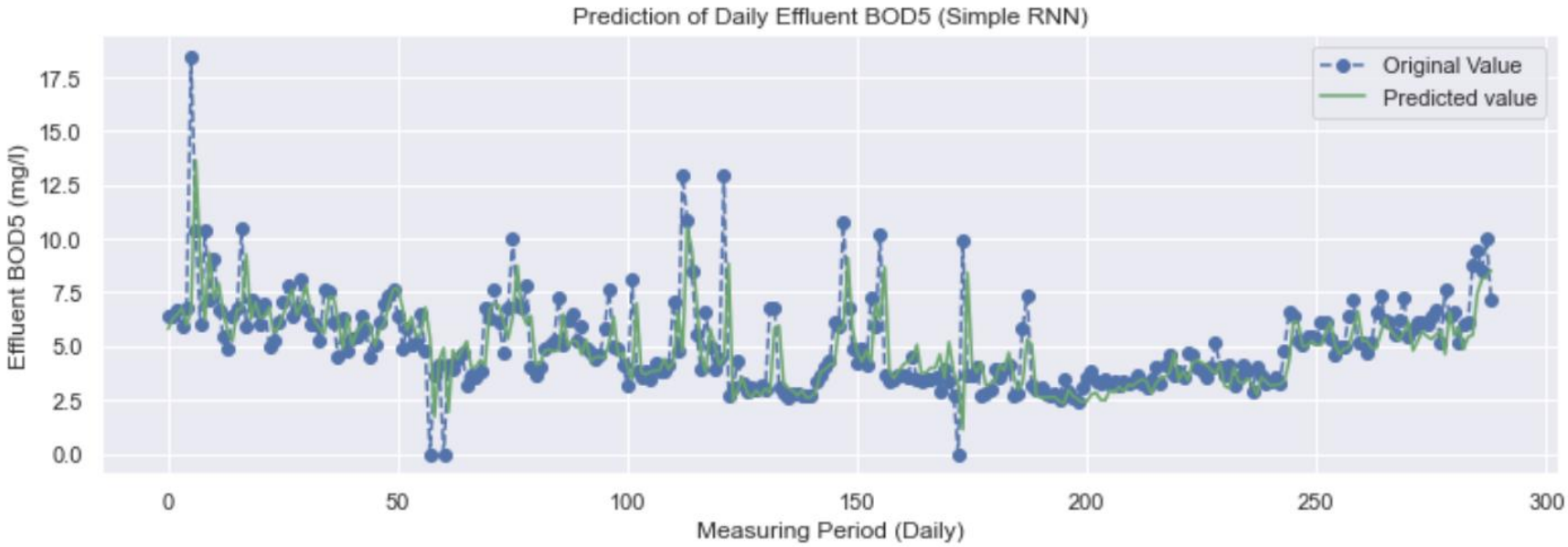


Figure 4.24 The prediction of daily effluent BOD_5 from 2015 to 2018 using the Simple RNN model.

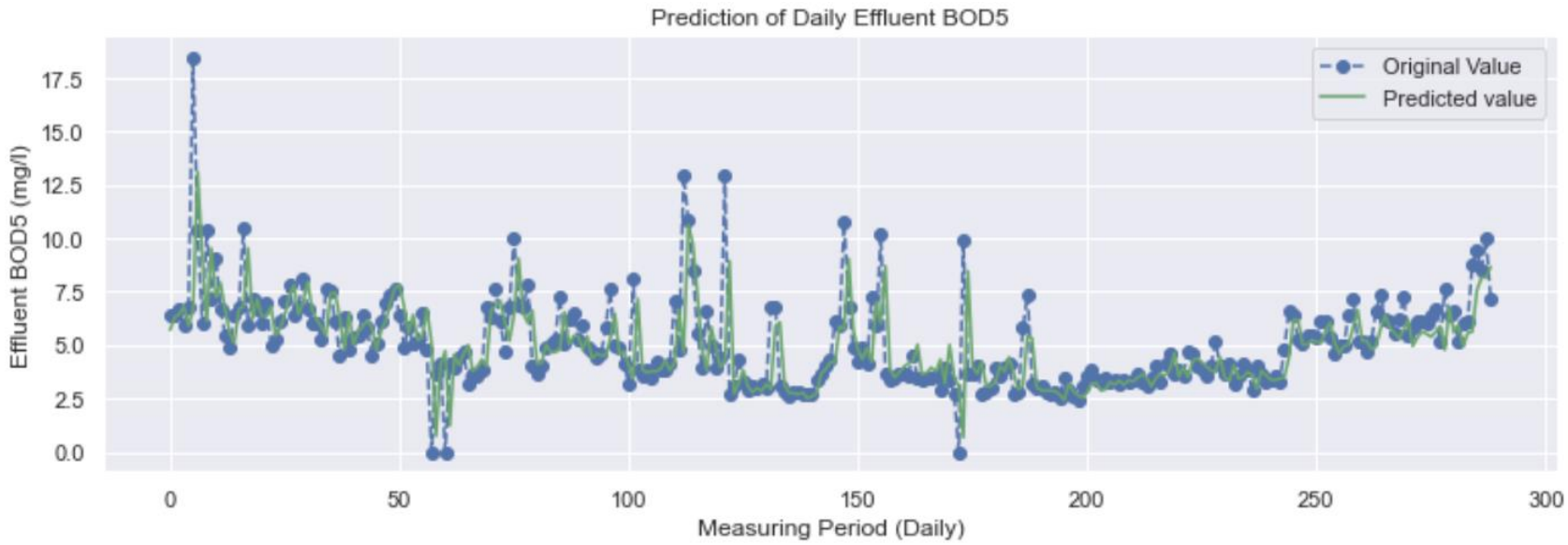


Figure 4.25 The prediction of daily effluent BOD_5 from 2015 to 2018 using the LSTM model.

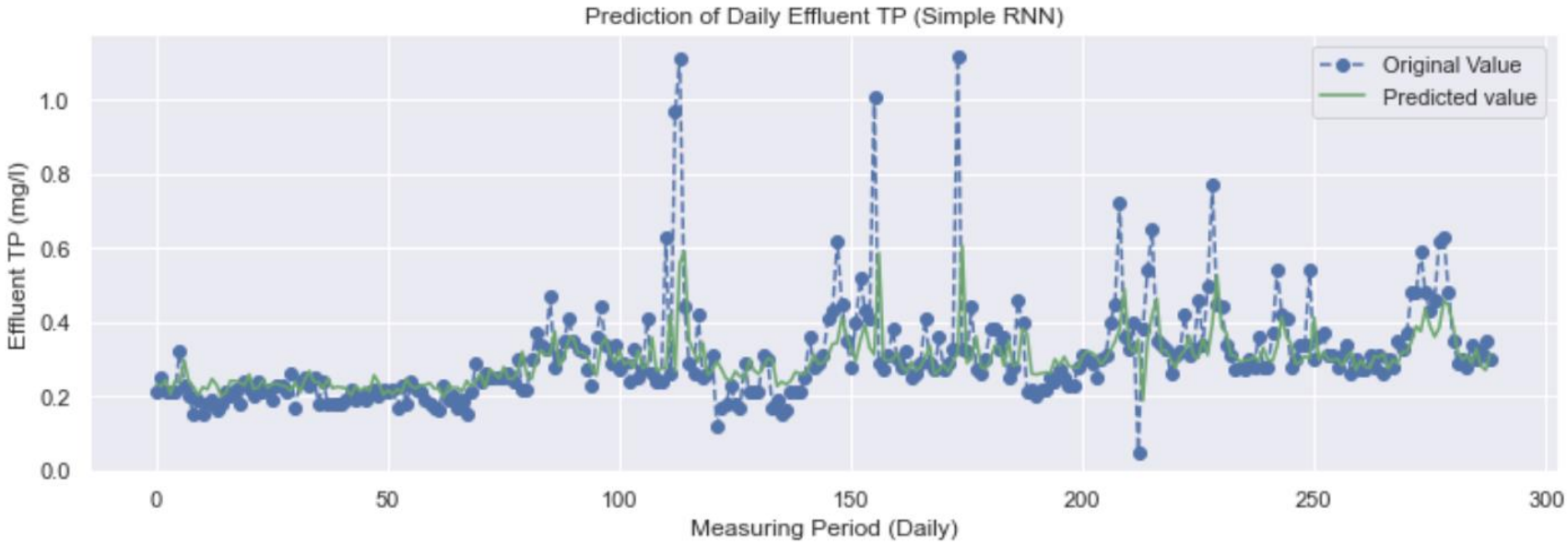


Figure 4.26 The prediction of daily effluent TP from 2015 to 2018 using the Simple RNN model.

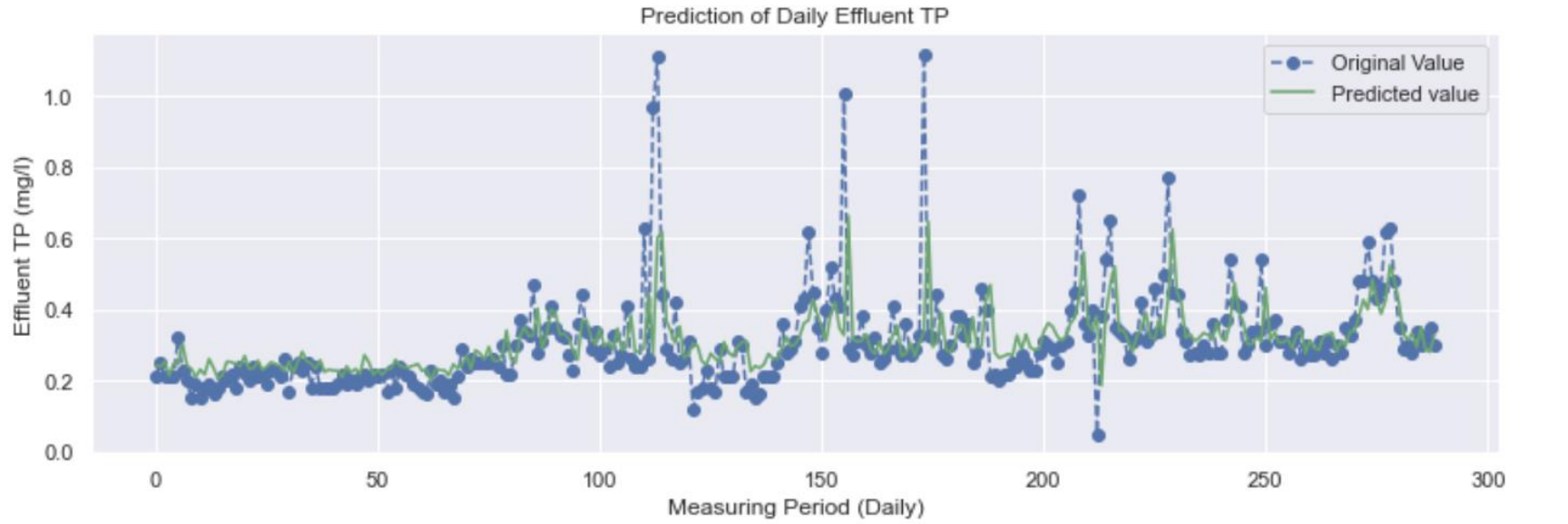


Figure 4.27 The prediction of daily effluent TP from 2015 to 2018 using the LSTM model.

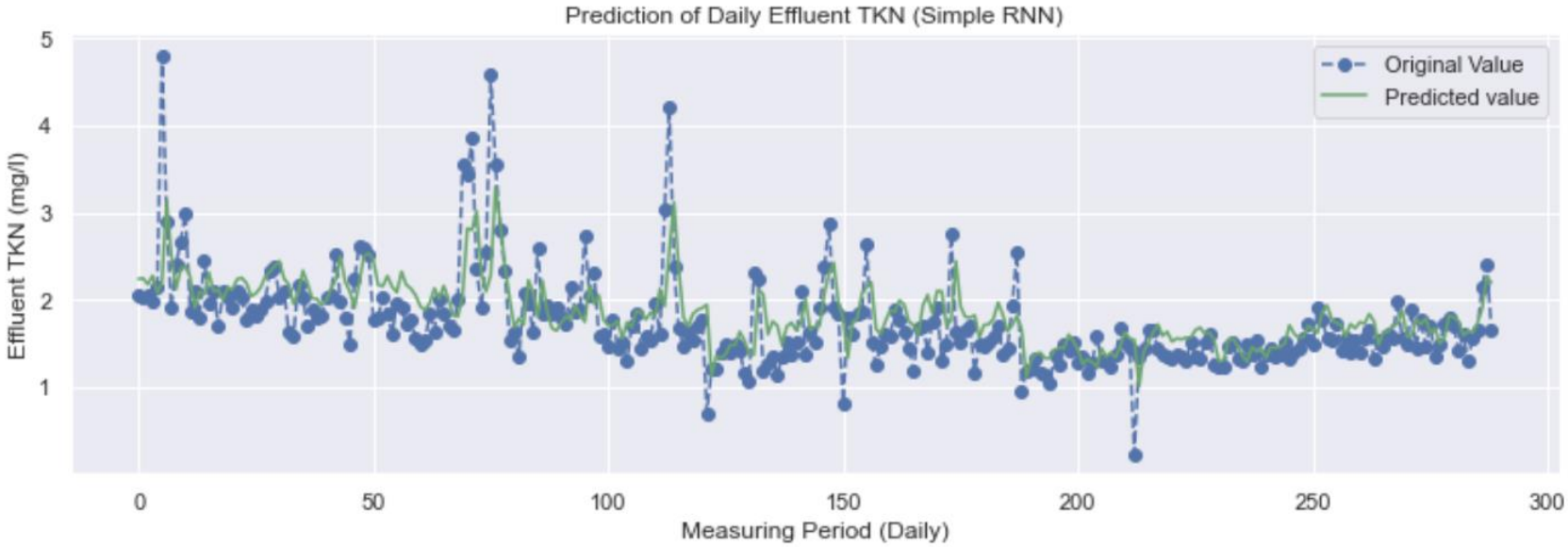


Figure 4.28 The prediction of daily effluent TKN from 2015 to 2018 using the Simple RNN model.

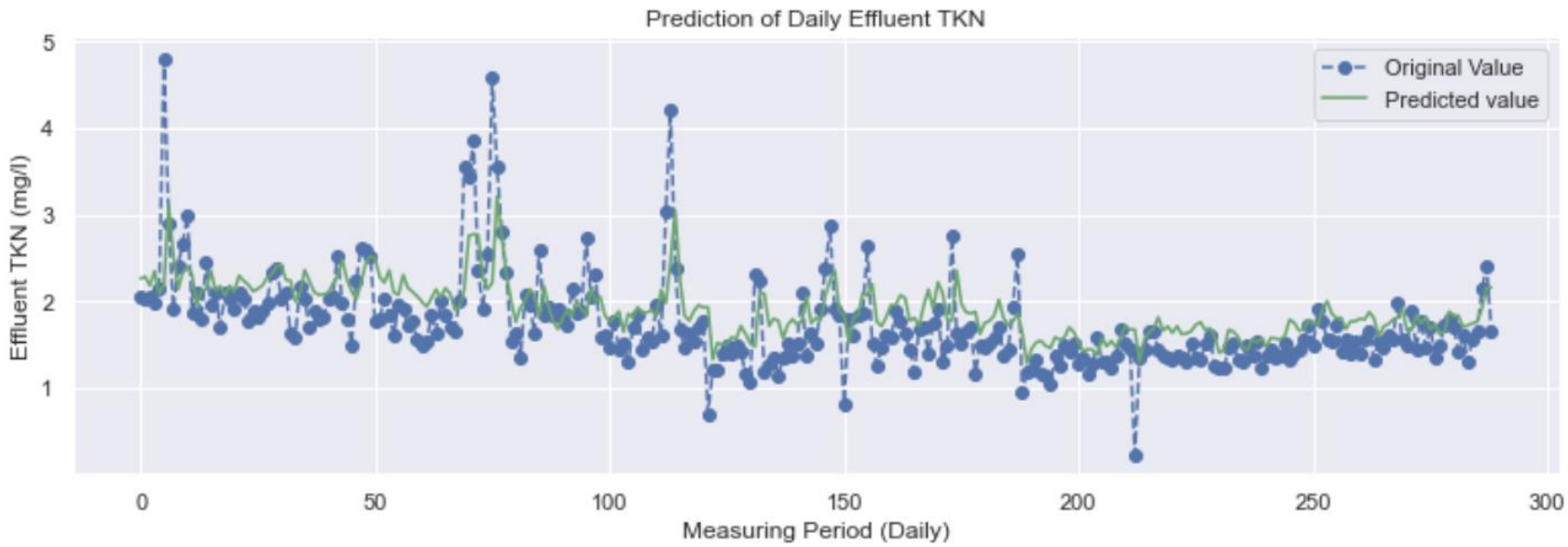


Figure 4.29 The prediction of daily effluent TKN from 2015 to 2018 using the LSTM model.

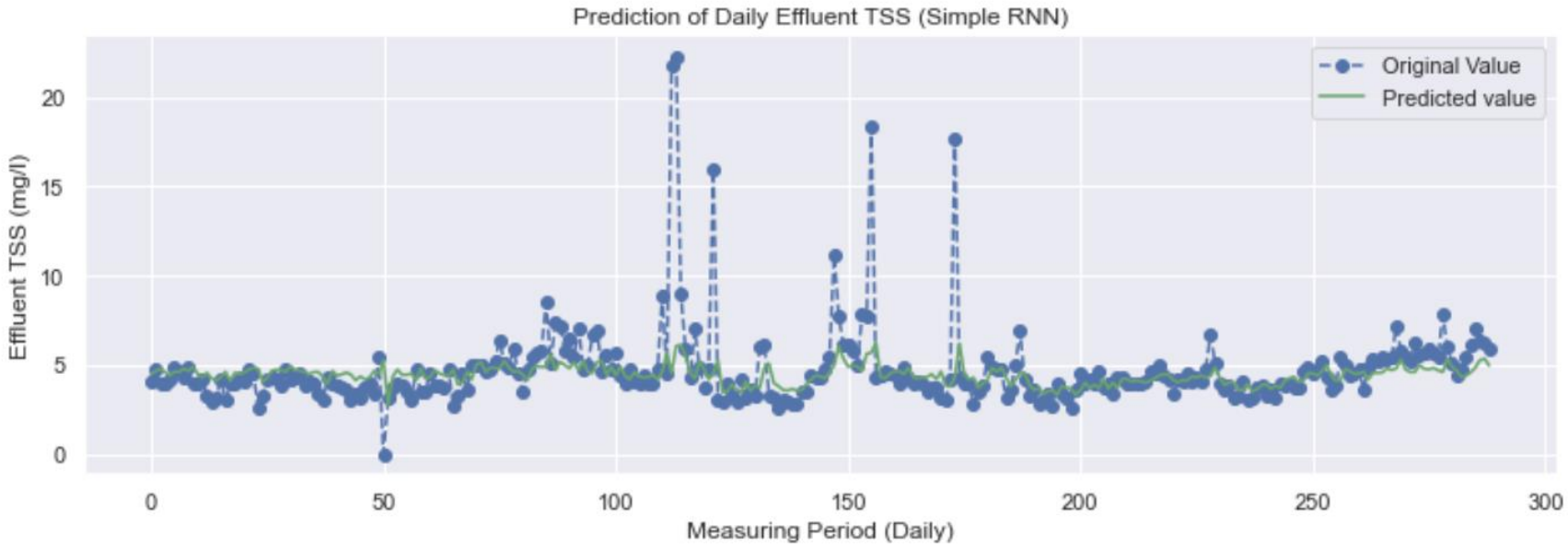


Figure 4.30 The prediction of daily effluent TSS from 2015 to 2018 using the Simple RNN model.

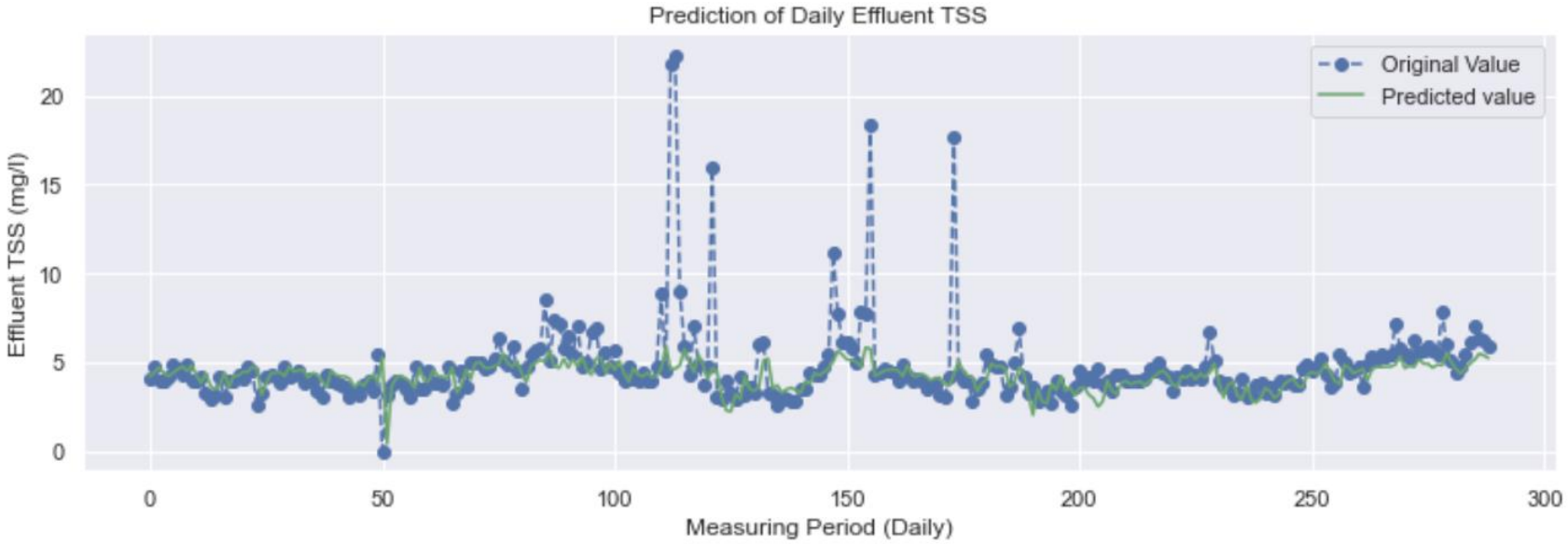


Figure 4.31 The prediction of daily effluent TSS from 2015 to 2018 using the LSTM model.

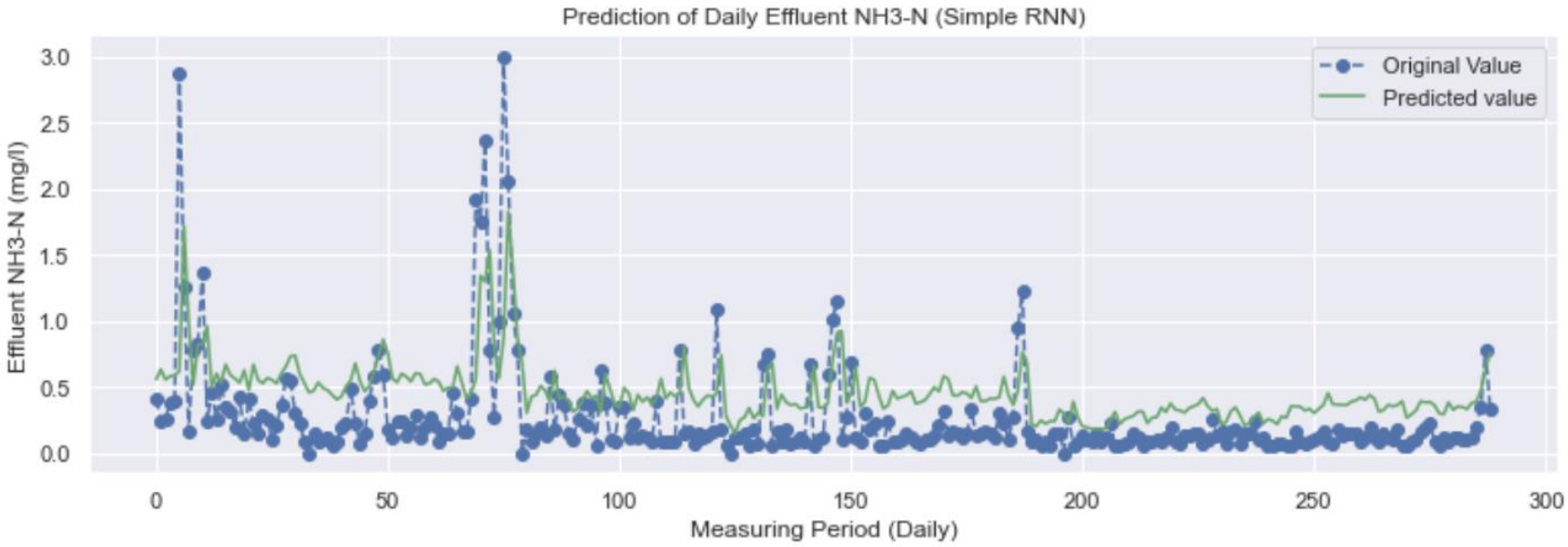


Figure 4.32 The prediction of daily effluent NH_3N from 2015 to 2018 using the Simple RNN model.

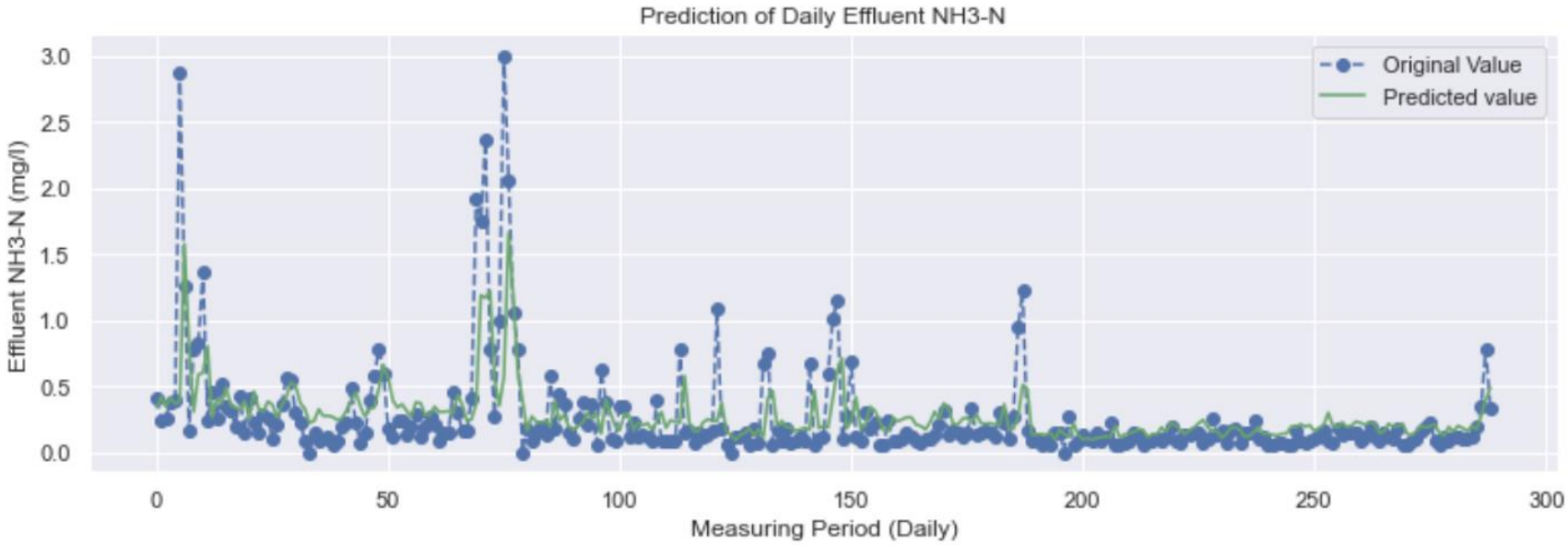


Figure 4.33 The prediction of daily effluent NH_3N from 2015 to 2018 using the LSTM model.

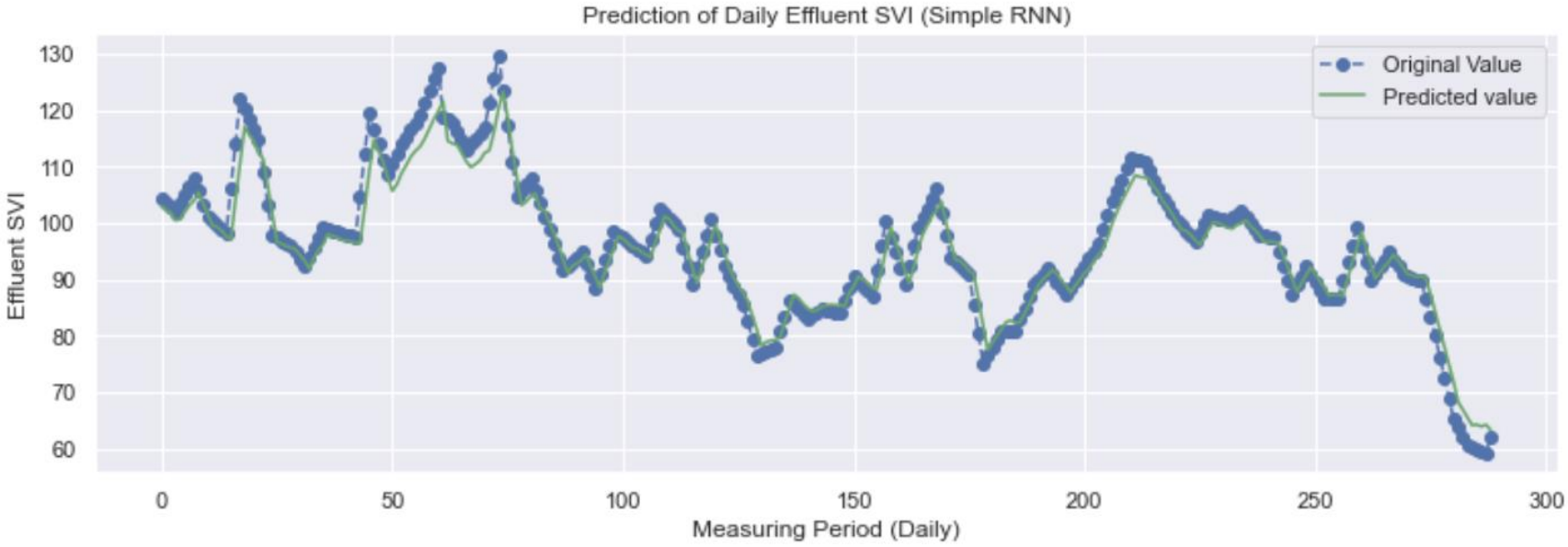


Figure 4.34 The prediction of daily effluent SVI from 2015 to 2018 using the Simple RNN model.

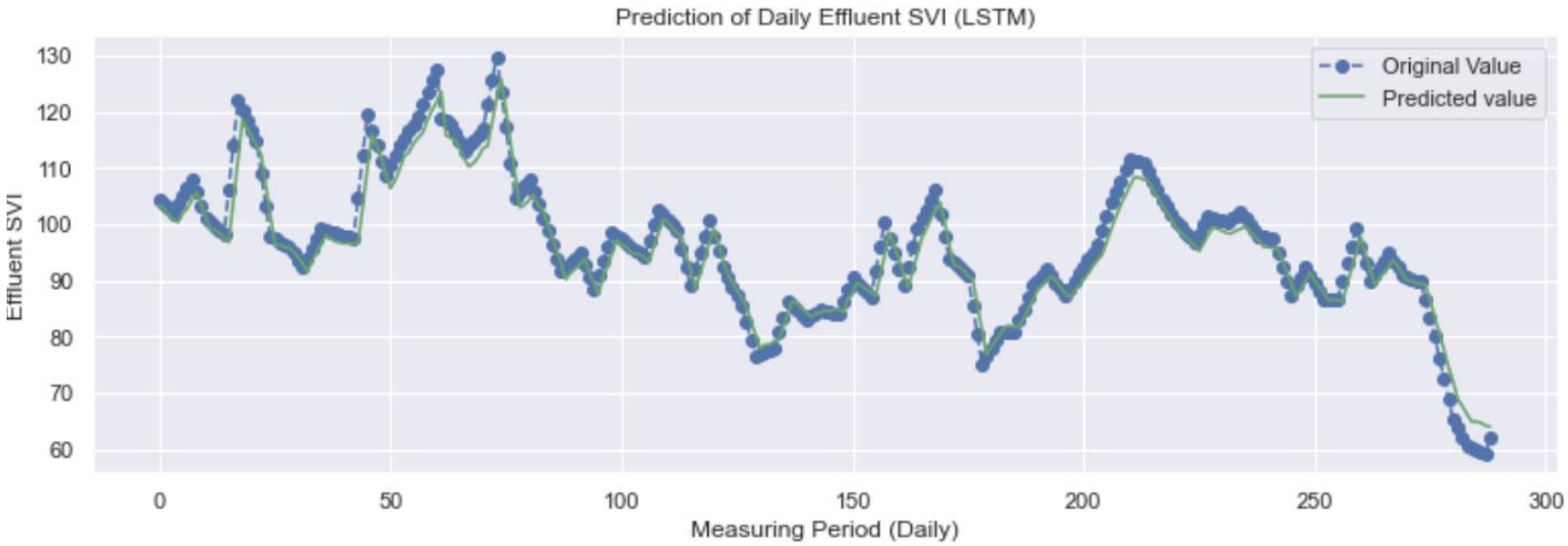


Figure 4.35 The prediction of daily effluent SVI from 2015 to 2018 using the LSTM model.

Table 4.7 compares the model accuracy between the discrete big data and the daily dataset for both Simple RNN and LSTM models. In addition, R^2 scores for the model evaluation are shown in Table 4.8. However, the R^2 score may not be appropriate for the time series model prediction (Hagquist & Stenbeck, 1998). It is suitable for linear regression. It is also can mean that the model is overfitting if the R^2 score is very close to 1.

Table 4.7 Comparison of the model accuracy between the discrete big data and the daily dataset.

Effluent Parameters	Discrete big data				Daily dataset			
	Simple RNN		LSTM		Simple RNN		LSTM	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
BOD_5	0.247	0.052	0.246	0.058	1.827	1.101	1.842	1.091
TP	0.019	0.004	0.018	0.003	0.117	0.066	0.117	0.069
TKN	0.091	0.066	0.067	0.027	0.427	0.293	0.448	0.328
TSS	0.319	0.051	0.320	0.074	2.178	0.988	2.212	0.975
NH_3N	0.054	0.025	0.056	0.029	0.364	0.284	0.311	0.169

The most appropriate metric to measure model accuracy is RMSE because it is the most suitable for a neural network model. RMSE interprets the same unit as the model, so the RMSE of the BOD_5 prediction for big dataset has the error between the predicted value and original value of 0.247 and the error for the daily dataset is 1.827.

Table 4.8 The average R2 score between the discrete big data and the daily dataset.

Effluent Parameters	R^2	
	Discrete big data	Daily dataset
BOD_5	0.985	0.260
TP	0.981	0.264
TKN	0.966	0.373
TSS	0.980	0.112
NH_3N	0.954	0.225

Poor prediction results are thought to be caused by a relatively smaller data size (1,400 sets). The previous model used more than 80,000 data. Of all the datasets, 80% were used for training and 20% for tests. Therefore, the daily prediction model may need a more significant number of datasets. A larger dataset can improve model accuracy, for example, the BOD_5 prediction has the RMSE of 1.827, and when applying a more extensive dataset, the model can achieve the RMSE of 0.247. However, for the daily SVI prediction, the result shows that the model has a good sample fit even though the dataset is small. Therefore, the variable data such as BOD_5 may need a larger training dataset for the model to learn better, but the stable data such as sludge volume index (SVI) may not need a very large training dataset.

The RMSE of the LSTM models is very similar to the simple RNN models. Figures 4.24 to 4.33 also show that prediction data is less accurate than the original big data prediction result. It should be noted that the prediction accuracy of the models will improve as more data is collected and used for training.

4.5 Conclusions and Recommendations

The Recurrent Neural Networks (RNNs) are excellent for developing a predictive model for sequential big data in WWTPs. The simple RNN and LSTM models were used to predict effluent parameters with different data sizes, epochs, and batch sizes. The following conclusions can be drawn:

- The prediction model was developed in five steps: define, compile, fit, and evaluate the network, and then make a prediction.
- Statistical analysis is one of the best tools to analyze, visualize, extract meaningful information. Meaningful data helps the model performs better results. As a result, scenario 2, data from 2015 to 2018, had excellent performance and short timing. In other words, a larger size of data collected from 1997 to 2018 did not have a better prediction because physical and operational conditions might have been changed.
- However, the daily dataset from 2015 to 2018 had less accuracy than the discrete big data model because of the number of training datasets.
- A large number of epochs and small batch sizes helped optimize the models in simple RNN and LSTM models with better timing and higher model accuracy, the RMSE value.
- The RMSE value of effluent BOD_5 prediction was better in the simple RNN models than the LSTM models.
- The fluctuation data was more suitable for the LSTM than the simple RNN model. In this dataset, BOD_5, TSS, and TKN fluctuated more than TP and NH_3N according to the standard deviation in Table 4.6. The results show that the LSTM model for BOD_5, TSS, and TKN had less RMSE and MAE than the simple RNN model for both big datasets and daily dataset predictions.

- Finally, the appropriate size of the dataset, an optimum epoch number, and optimum batch size are the most important factors to develop optimization models for WWTPs.

The simple RNN and LSTM models proposed in this study are robust and can be used to predict the wastewater treatment efficiency in WWTPs. This model development is expected to aid WWTPs in operation and process optimization.

4.6 Future Research Plans

Energy consumption can be controlled by monitoring dissolved oxygen (DO) and pump operation in WWTPs (Daw et al., 2012). With pumps, motors, and other equipment operating 24 hours a day, seven days a week, wastewater treatment facilities can be one of the largest consumers of energy in a community, and thus including the most significant contributors to the community's total greenhouse gas (GHG) emissions (U.S. EPA., 2013).

Therefore, it is recommended that the RNN and LSTM models be applied to predict DO and airflow rate to control energy consumption, enhance treatment operation, and ultimately protect public health and the environment.

5. LOGISTICS FOR A REAL-TIME PREDICTION MODEL

5.1 Abstract

As the increasing demand for clean water, security assessments and implementation of appropriate wastewater systems are received full attention. Many researchers have been exploring how the wastewater treatment system can be improved. With the development of technology, many predictive models have been developed. According to the previous studies, big data management, statistical analysis, and Artificial Intelligence (AI) to extract meaningful information, analyze historical data and develop a prediction model for WWTPs (WWTPs) using a Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) deep neural network algorithms. The model predicted accurately for important effluent parameters, Biochemical Oxygen Demand (BOD_5), Total Suspended Solids (TSS), NH3/Total Nitrogen (TN), and Total Phosphorus (TP) using a big dataset. The model also can precisely predict the daily SVI data because the algorithms can effectively handle sequential big data.

The traditional Supervisory Control and Data Acquisition (SCADA) system, which is monitored and controlled throughout the entire operation, has been widely used in most WWTPs. However, few predictive models feeding real-time wastewater treatment data exist. As the developed predictive algorithms, if the SCADA program is combined with Python, it can help optimize the plant by increasing the stability of the system, reducing risk beforehand, and finally decreasing operation costs. Moreover, Explanable AI can help to explain prediction values. It will make the new system to be trustable. In this study, real-time prediction logistics were developed. It will be

beneficial for WWTP operators to make better decisions in daily operation and detect potential problems in the operating performance beforehand.

5.2 Introduction

5.2.1 Background

The increasing demand for better control in WWTPs requires advanced technology for process optimization. Most WWTPs apply the SCADA system, a distributed computer system applied by operations and management for wastewater process monitoring and automation control. Wastewater treatment operators are constantly facing the challenge of finding the right balance between three main areas (Millinger, 2020):

- **Obtainability and reliability**: Including old infrastructure, stability of the system, and reliability of the information coming into the system;
- **Risk**: Compliance concerns, plant security, reporting and errors of the systems, and experienced operators retiring; and
- **Cost**: Operation and maintenance cost, chemicals cost, training new operators, and energy consumption costs.

By modernizing SCADA and Python, WWTPs can address challenges in these three important areas in several ways. Supervisors need to implement a secure-by-design modern SCADA that supplies information anytime, anywhere over a disaster recovery architecture for high availability and reliability. Moreover, the Explainable AI algorithm will make the system to be reliable. Risk can be reduced by reliable data management, effective data analysis, and consistent monitoring processes, as well as real-time prediction systems that can detect a potential risk beforehand.

Finally, more efficient operations can reduce cost by visualizing operational conditions, big data analysis, and optimizing operations by the Explainable AI.

5.2.2 SCADA Modernization with Python

There are automation technologies and computer software resolutions that can develop what is possible for wastewater treatment facilities. By modernizing existing supervisory control and data acquisition (SCADA) systems with Python programs, the complete solution enables the real-time SCADA to cooperate with the Python program to efficiently monitor and control the plant.

Python is a widespread high-level programming language known for its readability and performance. The operators can have high-performance visualization and real-time information, analyze the historical data, predict effluent to create fault alarm beforehand, and interpret the prediction values to make them trustable.

Modern SCADA with Python can help support a high level of service, eases compliance with developing regulatory standards, and enhances the efficiency of the field operators. Significantly, the plants can use the control layer as an organization for digital transformation to be prepared for the future. Due to SCADA's historically log process variables, Python will use the historical data to predict results and provide advanced warning to personnel when something seems amiss.

According to Millinger (2020), the author provides guides for SCADA modernization as follows:

1. Update the SCADA versions regularly.

Updating the system is the first step. Many WWTPs are still on old versions of SCADA software. Regular updates and alignment to the latest versions improve system accessibility, while a delay in updates can raise security concerns. Therefore, the plants need to make sure that the SCADA and operating system are the latest before taking other steps to modernize solutions involving analytics or web-based interfaces.

2. Standardize the SCADA system.

Enhance competence by defining standards for the overall SCADA systems, including application, configuration, security, architecture, and remote access. Standardization will help lessen faults, reduce costs, improve operations efficiency, and ensure compliance. Lastly, plant operators can control sources and automatically build a SCADA process database with tag name conventions.

3. Implement data management.

The correct data is essential. A plant cannot operate successfully without good data management. Reliable data with effective data management from different sources can allow the facility to expand and scale as systems develop. Current technologies make information available to stakeholders who are not directly connected with the SCADA but need data to make the right decisions, such as demand, planning, and forecasting.

4. Create proactive alarm detection.

Good alarm technology can reduce noise, faster reactions, improved productivity and efficiency, and safer operations to immediately move from an alarm to notification and guide the right action. Moreover, better alarm detection can handle errors occurring in facilities in advance. This can help avoid system failure and make safer wastewater treatment systems.

5. Digitize work processes.

Every WWTP has standard operating systems but mostly printed manuals. At this point, a plant can move from manuals to incorporating work procedures into its modern SCADA system. Using SCADA data, management can activate a work process, lead operators through steps and improve operational consistency.

6. Drive business connectivity.

Through the entire wastewater system, SCADA connectivity offers a complete view of system performance, fills data gaps, and enlarges cooperation. Centralized data management drives constancy across plants and sites. The modern SCADA will improve the data management system by leveraging secure-by-design thin clients on low-cost hardware to make data readily accessible to all levels of the organization. The data will be a persona-based visualization by providing each worker with the information and capabilities they need, rather than the onsite SCADA screen. The system can help the workforce with customized information and remote monitoring/control abilities, saving operator time and speeding up the response and compliance.

These are essential guidelines for making the SCADA modernization in wastewater treatment facilities. Lastly, enable prediction model-based in Python program can leverage industry standards to map a prediction model to an equipment model by organizing data and providing standard context across locations and data sources. Operators can quickly navigate in context with information derived from the model. Modern SCADA enables a user experience regardless of the monitor, device, equipment, role, or process. A SCADA system with the Python program can guide controllers to operate WWTPs. As supervisors enable remote monitoring within the plant, they will experience greater competence because modern systems connect machines, data, insights, and people

Implementing high-performance SCADA with Python can increase operator efficiency through screens with proactive information. This implementation improves situational awareness, fault detection, and productivity whereas decreasing the risk of errors. Today’s SCADA is not just monitoring and visualization with alarms rolling in. It is about optimizing operations for proactive decision support. With the latest innovations in designs for efficiency, operators can quickly identify problems and causes for a fast resolution.

5.2.3 Explainable Artificial Intelligence

As predictive algorithms play an important role in our lives, they become increasingly complex. Explaining why an algorithm makes certain decisions is ever more crucial (Ghorbani et al., 2019). Accuracy and interpretability are two main factors of successful predictive models. Typically, a decision must be made in favor of complex black box models such as Recurrent Neural Networks (RNN) for accuracy versus less accurate but more interpretable traditional models such as the logistics regression model (Choi et al., 2017). However, the highest accuracy for big datasets is often attained by complex models that experts struggle to explain. Several approaches have been developed to help users understand the predictions of complex models. Still, it is often unclear how these approaches are related and when one technique is preferable over another (Lundberg & Lee, 2017).

According to Lundberg & Lee (2017), the authors presented a method for interpreting predictive models called SHAP, SHapley Additive exPlanations. SHAP assigns each feature a weight value for a particular prediction. The SHapley values of a conditional expectation function of the primary model; thus, they are the solution to Equation 5.1 (Lundberg & Lee, 2017).

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{5.1}$$

where $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z'vector where the non-zero entries are a subset of the non-zero entries in x'

SHAP (SHapley Additive exPlanation) values are a unified measure of feature importance (Lundberg & Lee, 2017). In other words, SHAP assigns each input/feature an importance value for a particular prediction. They interpret how to get from the base value E[f(z)] that would be forecasted when the unknown featured to the current output f(x) occurred. Figure 5.1 shows the

diagram for a single ordering. The SHAP values start from averaging the -i values through all possible orderings. The calculation of SHAP values determine the importance of a feature by comparing what a model predicts with and without feature. The calculation is made in every possible order because the order of a model can affect a prediction. Therefore, it is suitable for a non-linear or complex time-series data, which the order of data affects a prediction output.

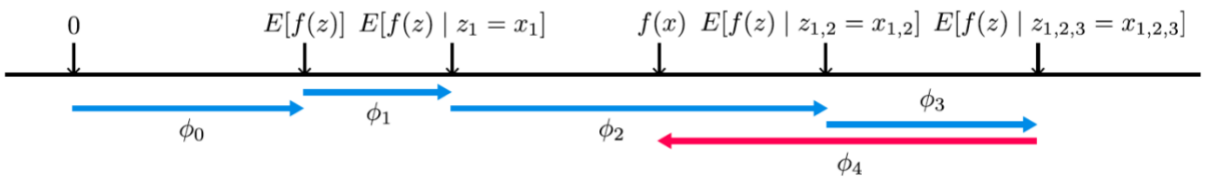


Figure 5.1 The diagram that shows a single ordering. (Source: Lundberg & Lee, 2017)

5.2.4 Literature Review of Aeration Optimization in Wastewater Treatment Process

There have been ongoing trials of saving energy in the activated sludge wastewater treatment process. The first method of reducing energy is to improve the aeration system for maximum oxygen transfer efficiency. The oxygen transfer rate can be significantly improved by replacing coarse bubble diffusers or surface aerators with fine-pore diffusers (Rosso et al., 2008). The second method is to select energy-efficient blowers. The last most challenging approach is to control the aeration in real-time to minimize energy consumption. Figure 5.2 shows the three steps for aeration optimization.

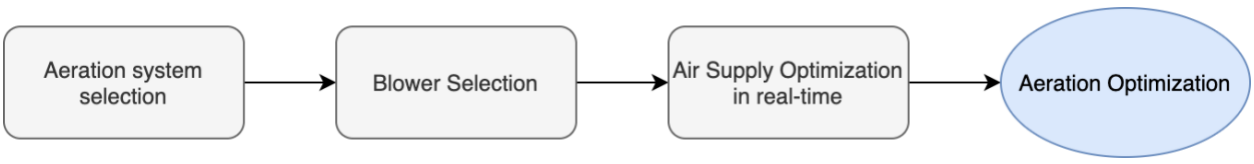


Figure 5.2 Three steps of an aeration optimization method.

Selecting a suitable aeration system is crucial. There are several components in the aeration system that affect the operation of the aeration blowers. For example, the depth of submergence controls the

hydrostatic head that the blowers must overcome. The submersion in a typical municipal WWTP ranges typically from 3 m (10 feet) to 7.9 m (26 feet), which requires aeration blowers pressure under 1.05 kg/cm (15 psig). In contrast, the industrial system requires 10 m (33 ft) to 20 m (66 ft) of submersion (Aerzen, 2021). Therefore, selecting a system and its components is essential to optimize the energy in an aeration system.

5.2.5 Study Objectives

The objectives of the study are to:

- Improve the traditional SCADA system combining with Python program to apply the predictive models for a real-time wastewater treatment system;
- Apply an Explanable Artificial Intelligence approach to interpret the predictive values to enhance the model reliability; and
- Develop real-time logistics for a WWTP operation.

5.3 Materials and Methods

5.3.1 Logistics for a Real-Time Model in Wastewater Treatment Plant

A supervision control and data acquisition (SCADA) system simulator is a Human Machine Interface-based software that allows visualizing the process of a plant. Madison Metropolitan Sewerage District (MMSD) has applied the SCADA system to monitor and control the current plant operating conditions. According to the previous studies, the deep neural network models can effectively predict the effluent result. Therefore, a dynamic simulation of effluent prediction models will significantly benefit a WWTP. Developing the logistics of the SCADA system, applying the deep learning model will facilitate operators in monitoring, operating, handling the alarm, retrieving historical data, and historical trend in the WWTP. It will be extremely advantageous for plant operators to identify fault errors in the system operation beforehand.

The first step in this task is to conduct a literature study from several reliable resources and review the SCADA operation system in the Nine Springs WWTP. A SCADA system is a communication and control system applied to monitor, operate, and maintain infrastructure on the network and acquire data (Wu et al., 2006). SCADA is widely used in large to medium-sized plants (Morsi et al., 2009). The general architecture of the SCADA system is shown in Figure 5.3 (Sosik, 2014).

The SCADA System's components consist of one or more CPU's (Central Processing Units), RTU's (Radio Telemetry or Remote Terminal Units), I/O subsystems, video monitors, field sensors, control devices, and various software that drives the I/O, runs the control algorithms, creates control outputs, presents graphics and monitored values, detects alarm statuses, and keeps the monitored points in a series of data files that can be recalled later for analysis or process verification (Sosik, 2014).

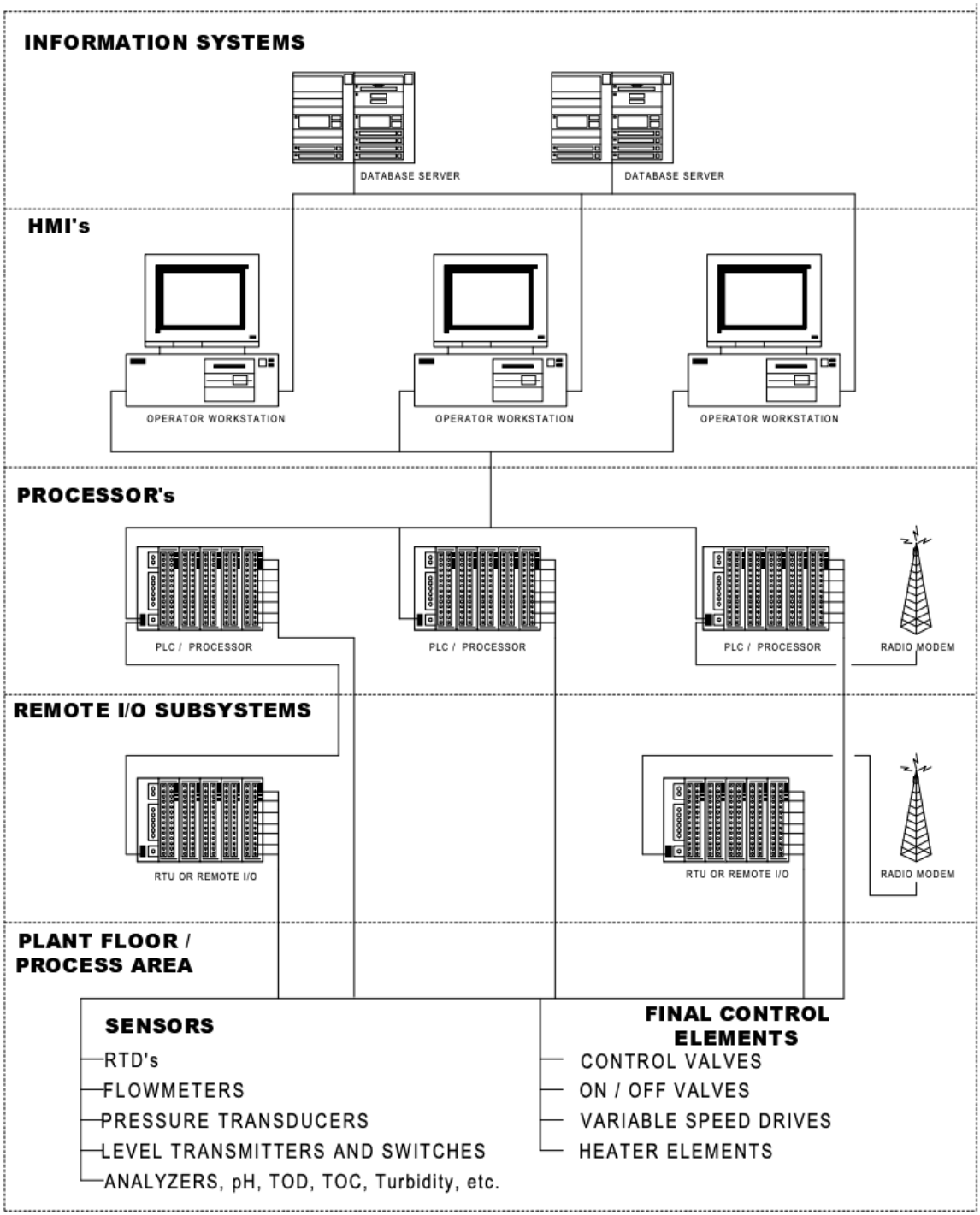


Figure 5.3 The general architecture of the SCADA system. (Source: Sosik, 2014)

Figure 5.4 shows the main components of the SCADA system consisting of HMI, SCADA master, RTU, PLC, communication channels, and physical devices such as sensors, meters, and other

appliances (Manda et al., 2018). HMI (Human Machine Interface) is a device that displays the data collected from operational processes to operators. It assists operators in running the devices remotely by sending the instruction to machines. The SCADA master is a computer server. It gathers, stores, and handles data from all devices and then sends back the control commands.

Remote Terminal Unit (RTU) and Intelligent Electric Devices (IED) interact between the SCADA and devices. They convert sensor signs to data and transfer the data to SCADA master. Programmable logic controllers or PLCs are an economical option for RTU. PLCs are used as field devices. They are flexible and multi-purpose and can be arranged when compared to RTU. Furthermore, communication channels can connect the SCADA master to the remote units. Lastly, physical devices are sensors, meters, and other pieces of equipment.

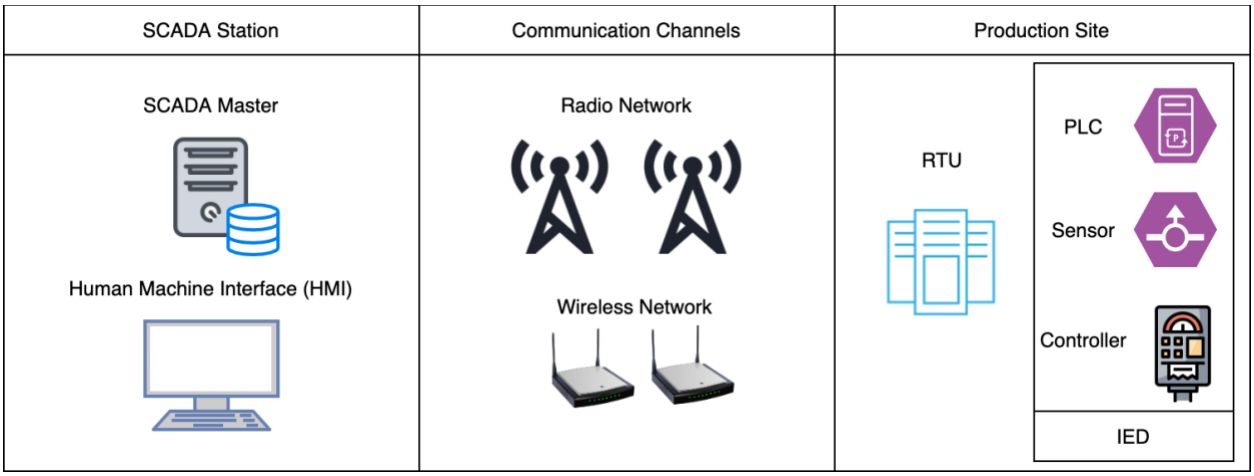


Figure 5.4 The main components of the SCADA system. (Modified from Manda et al., 2018)

The second step is to develop a schematic by inputting the components existing in the previous study. The third step is to determine the relationship of each element in the plant to identify the working principle of the linkage. Figure 5.5 shows a general layout of a WWTP having different inputs for each process (Richards, 2020).

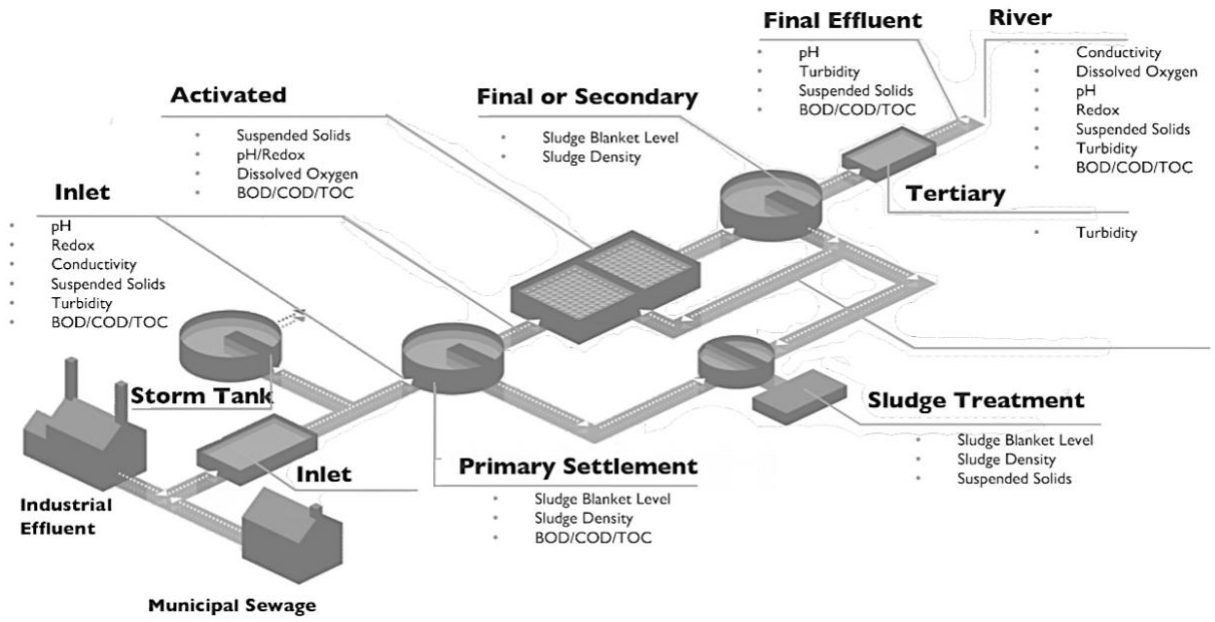


Figure 5.5 General WWTP layout. (Source: Richards, 2020)

After determining the relationship of each component, the fourth step is to create a collection scheme for the operations. The fifth step is to develop historical trends to display stored past data graphically. The sixth step is to apply the model predictions and develop an explainable artificial intelligence for interpreting the prediction value. Finally, the last step is to present the logistics of real-time schematics starting from the historical trend, model prediction, explainable model, and finding the operation optimization.

The overview of this research is presented in Figure 5.6. The previous research includes big data management and effluent model prediction. In this study, logistics for the real-time system was developed. The capabilities and essential features of the developed system will benefit the economy, social health, and the environment.

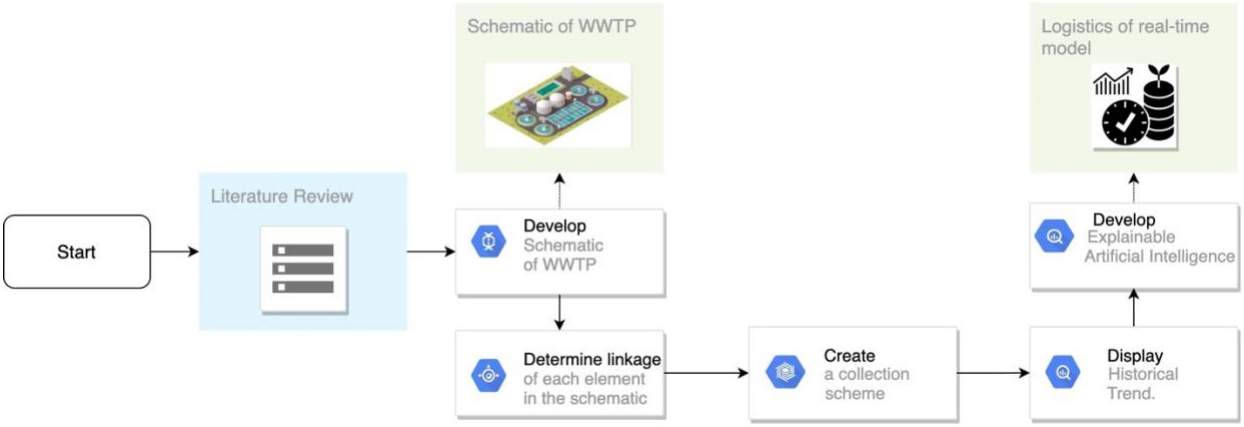


Figure 5.6 The flow chart of the real-time logistics system.

Data Collection from Wastewater Treatment Plant

After the literature review, the data collection points diagram in a WWTP was created in Figure 5.7. Moreover, in Figure 5.8, the main automation control of each system has been studied (Du et al., 2019). The main points of energy consumption and chemical consumption were evaluated. To develop its modern SCADA system, WWTPs need to create the data collection scheme and the main point of energy and chemical controls.

The data is collected between each system. Start from influent entering to the plant. Pumps bring wastewater from the wet well, and one main energy consumption happens here. Influent data is collected, and then the wastewater is sent to preliminary treatment to remove inorganic substances. After that, the effluent from the preliminary treatment will be sent to primary treatment to remove settleable inorganic matters. Chemical consumption happens at this stage. The data is also collected. Then, primary treated sewage is sent to secondary treatment to remove organic matter. In this process, biological treatment will occur. Air is added to this process, so this is the main point of energy consumption. Then, the tertiary stage is disinfection. The main chemical consumption occurs in this step because the chemical added to treat water, such as chlorine.

However, some plants use ultraviolet (UV) disinfection that is a leading energy consumption source.

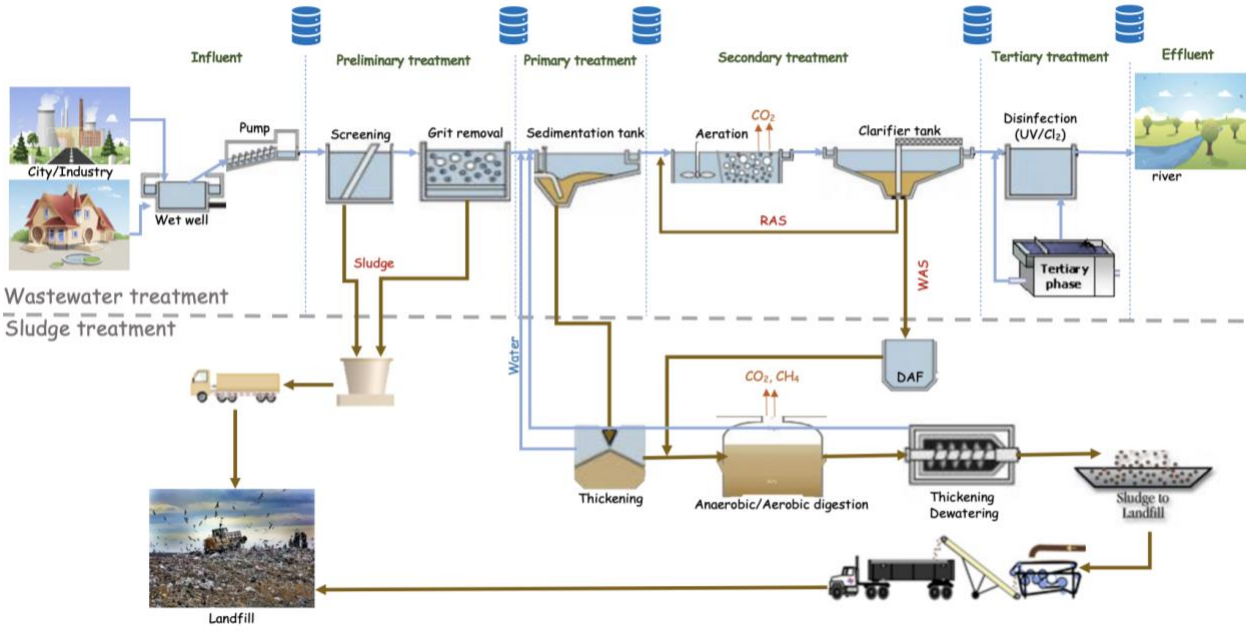


Figure 5.7 The data collection diagram in a wastewater treatment facility.

Figure 5.9 shows the main components of the modern SCADA with Python. The SCADA system with Python program will enhance wastewater treatment by creating data visualization, model prediction, and system alert. Then, the communication channels will send out the commands to the production site to control the systems.

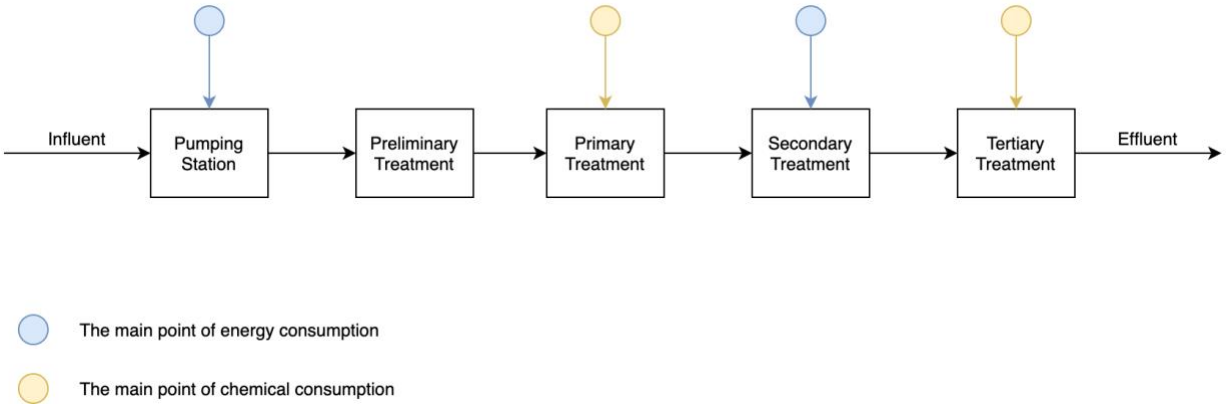


Figure 5.8 The automation control system of a WWTP. (Source: Du et al., 2019)

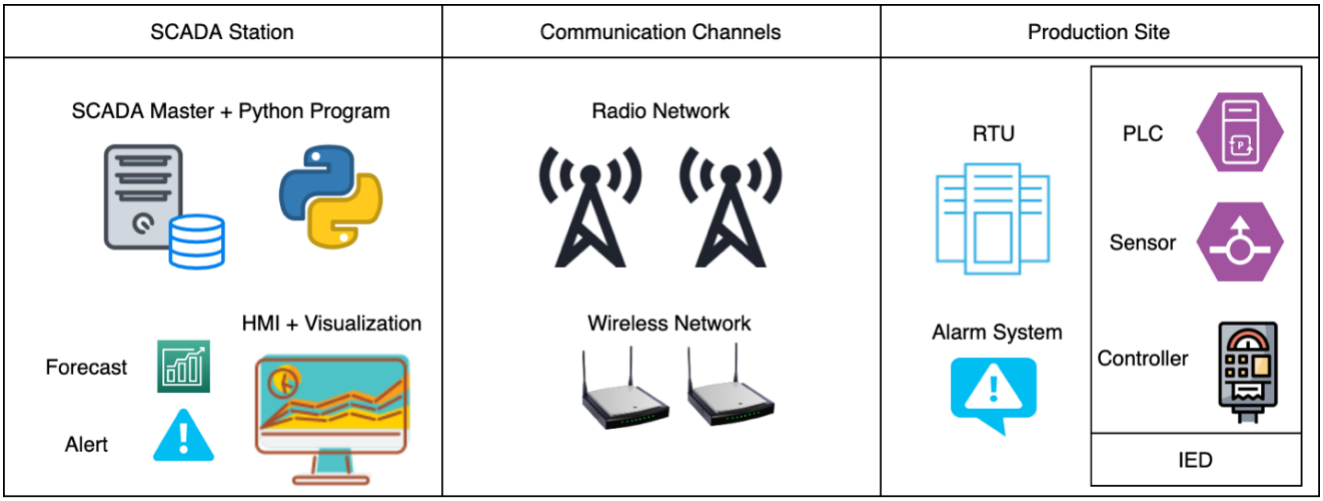


Figure 5.9 The structure of the modern SCADA. (Modified from Manda et al., 2018)

Figure 5.10 shows the application of the modern SCADA to WWTPs. The modernization system can provide online monitoring, data analysis, system alerts/control, and real-time/proactive operation and maintenance. The traditional SCADA can mainly monitor, control a wastewater treatment system. However, the modern SCADA will enhance the capability of the conventional SCADA. The program will not only monitor and control, but it will combine the prediction model into the system

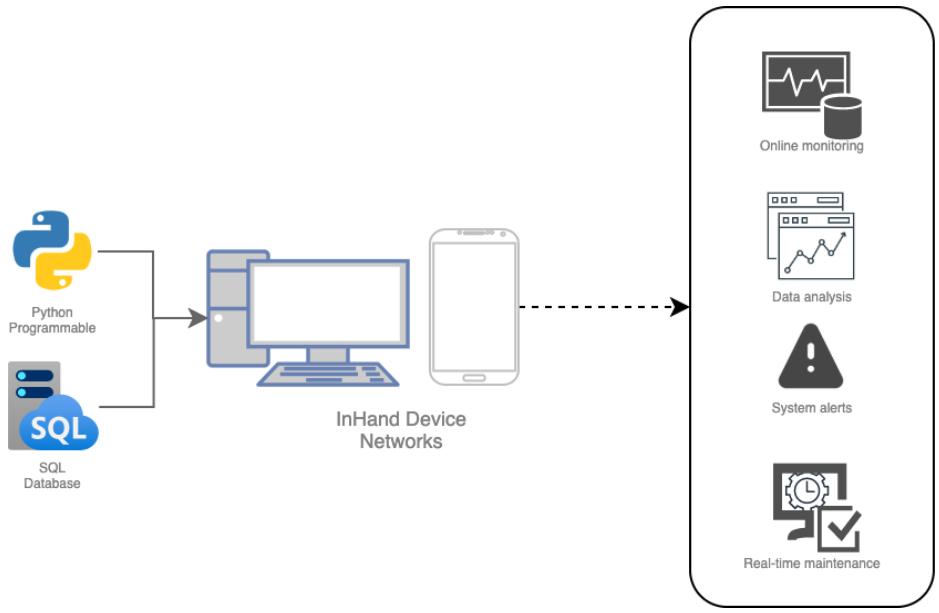


Figure 5.10 The modern SCADA with Python.

5.4 Results and Discussion

Recurrent neural network (RNN) models were used to predict effluent parameters. The Explainable AI algorithm was also applied to determine which parameters affect the outcomes most by the function called SHAP (SHapley Additive exPlanations) in the Python program. First, SHAP needs to be imported into the program and then executed to obtain the predicted result. The model predicts effluent BOD_5 using the RNN algorithms. The input values were the influent parameters, BOD_5, TP, TKN, TSS, NH_3N, and flow rate. Organic loading values are calculated from flow rate and influent BOD_5. The output is effluent BOD_5. Table 5.1 shows the dataset from 1996 to 2020, including output and input parameters.

Table 5.1 The dataset of the model including output and input parameters.

Date	Effluent_BOD5	OrganicLoading	Flow_In (MGD)	Influent_BOD5	Influent_BOD5	Influent_TSS	Influent_TKN	Influent_NH3-N	Influent_TP
1996-07-29	4.0	8685.6000	41.3600	210.0	210.0	200.0	27.0	16.9	6.13
1996-07-30	4.5	9177.2100	41.6200	220.5	220.5	215.0	28.2	16.5	6.56
1996-07-31	5.0	8813.7150	40.7100	216.5	216.5	245.0	28.8	17.5	6.98
1996-08-01	4.8	9510.3700	41.5300	229.0	229.0	216.0	28.4	18.4	7.33
1996-08-02	4.5	7900.2000	41.0400	192.5	192.5	166.0	27.7	18.8	6.32
...	...	...	...	...	...	...	...	...	...
2020-03-02	7.4	9645.4342	42.1198	229.0	229.0	247.0	42.8	28.1	5.38
2020-03-03	6.9	9231.5508	42.1532	219.0	219.0	75.6	41.3	29.4	4.30
2020-03-04	6.9	9272.0877	42.3383	219.0	219.0	197.0	47.2	31.1	5.36
2020-03-05	6.9	9330.5826	42.6054	219.0	219.0	197.0	47.2	32.7	5.36
2020-03-05	6.9	9330.5826	42.6054	219.0	219.0	197.0	47.2	32.7	5.36

8642 rows × 9 columns

Figures 5.11 to 5.13 show the summary of the SHapley values in a plot with different inputs. Each point on the summary plot is a SHAP value for one observation and input. The plot ranks variables in descending order on the y-axis, which is called feature importance. The horizontal location shows the impact of that value associated with a higher or lower prediction. Figure 5.11 shows that NH_3N had the highest impact on the effluent BOD_5, followed by BOD_5, flow rate, TKN, TSS, and TP. Figure 5.12 also shows that NH_3N has the most impact on the effluent BOD_5, followed by TKN, organic loading, TP, and TSS. Figure 5.13 shows the highest impact of NH_3N on the prediction values. Therefore, the plots of all models imply that NH_3N had a high positive impact on the effluent BOD_5. The red color means that higher influent NH_3N increased the predicted effluent BOD_5. The blue color implies that the lower influent NH_3N decreased the predicted values. The same as the higher of influent TKN and BOD_5 increased predicted effluent BOD_5. Unlike influent TSS and TP, the increase of influent TSS and TP lowered the predicted effluent values. Therefore, the plot indicates that TSS and TP were negatively correlated with the target variable.

The finding also demonstrates that real-time ammonia monitoring was suitable for controlling the air supply to aeration basins but not sufficient for better control of aeration basins.

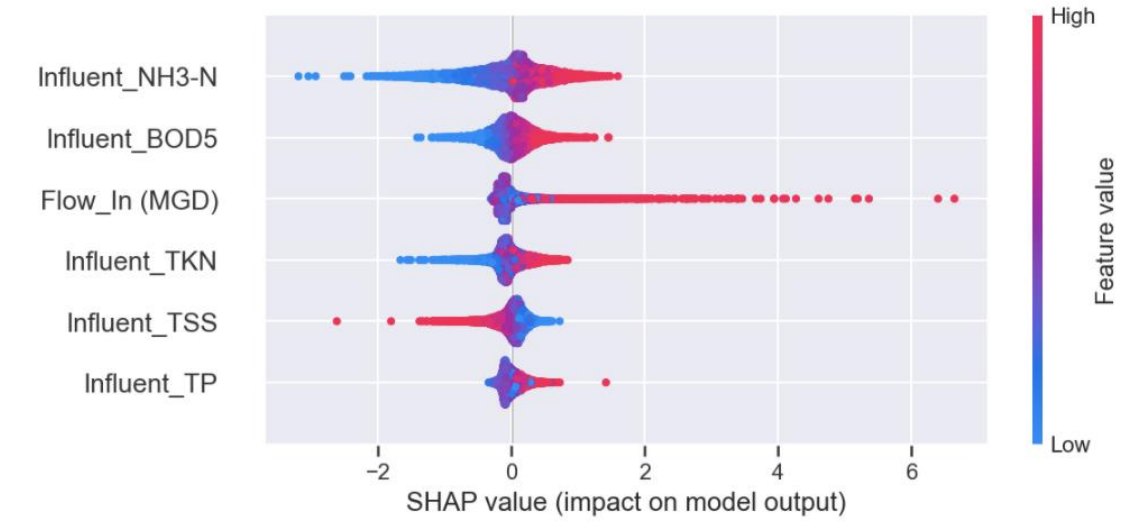


Figure 5.11 The SHapley summary plot of the model that includes flow rate.

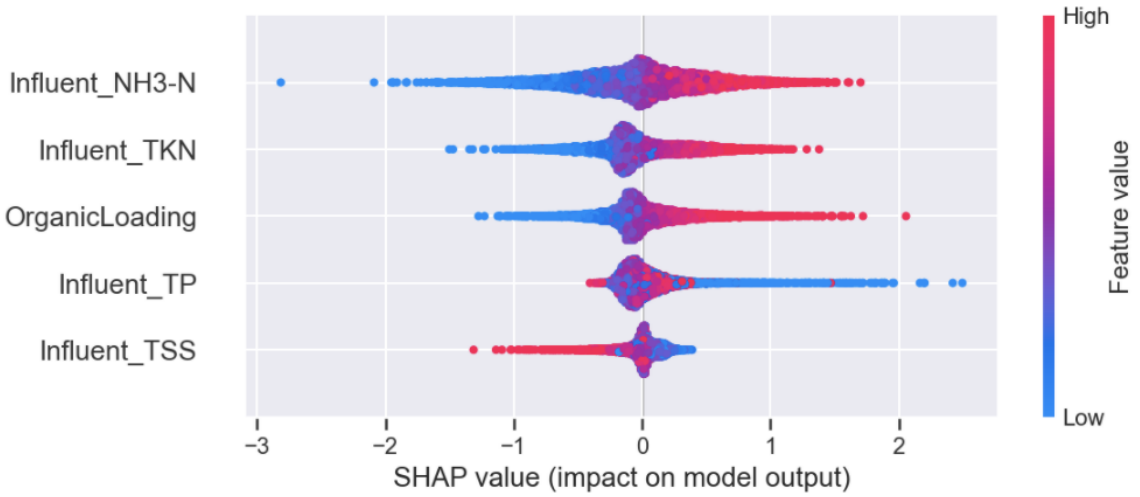


Figure 5.12 The SHapley summary plot of the model that includes organic loading.

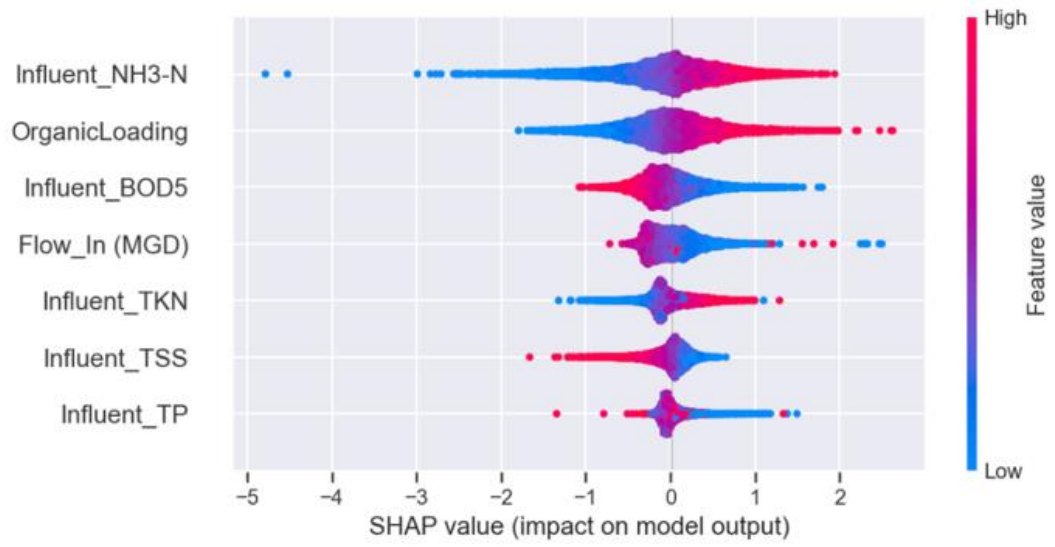


Figure 5.13 The SHapley summary plot of the model that includes flow rate and organic loading.

Figures 5.14 to 5.16 show the force plot of the first observation in the three datasets. The f(x) value in the figure is the effluent BOD_5 range. Figure 5.14 shows the first observation of the prediction of effluent BOD_5, which is 2.15 mg/L. The result reveals that parameters that drove the predicted effluent BOD_5 lower (blue color) were NH_3N, TKN, BOD_5, flow rate, and TP in descending order. In contrast, TSS predicted higher effluent BOD_5 (red color). The figure also shows each value of input parameters in this observation.

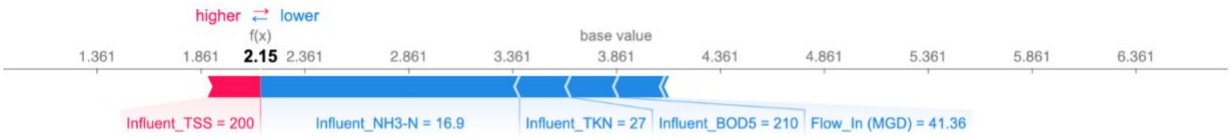


Figure 5.14 The force plot of the first observation including flow rate.

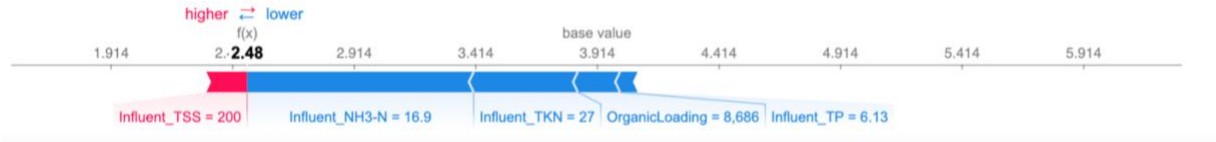


Figure 5.15 The force plot of the first observation including organic loading.

Figure 5.16 The force plot of the first observation including flow rate and organic loading.

Each observation has its own force plot. Figure 5.17 shows the combination of all plots by rotating 90 degrees and stacking horizontally. The x-axis of the above figure is the y-axis of the individual force plot. This plot shows the first ten observations. The figure shows that the influent NH_3N was the most related to the effluent BOD_5 in this first ten observations because influent NH_3N has the widest band in the graph. Therefore, the most substantial influence of the influent NH_3H on the predicted effluent BOD_5 implies the importance of oxygen supply to aeration basins because the removals of BOD_5 and NH_3H are affected by the dissolved oxygen (DO) level in aeration basins.

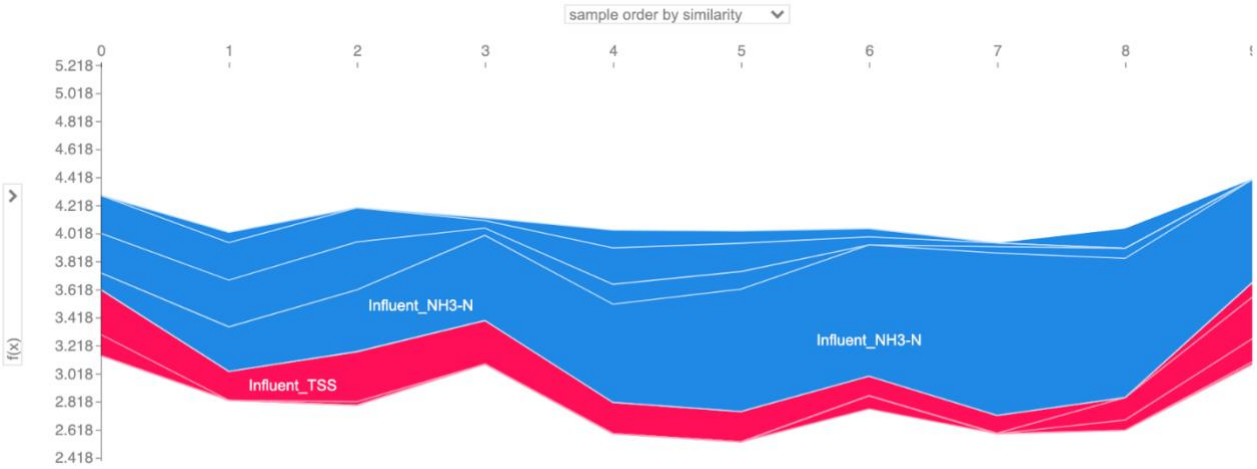


Figure 5.17 The collective force plot of all input variables.

Figure 5.18 shows the multiple input values in the collective force plot by selecting by similarity with different input values. The first model included the flow rate as input when the effluent BOD_5 was 3.336 (on the y-axis), the influent TP, TSS, NH_3N, and BOD_5 concentrations were 6.32, 166, 18.8, and 192.5 mg/L, and the flow rate was 41.04 million gallons per day (mgd). The second model included organic loading as input when the effluent BOD_5 was 2.476, and the influent TSS,

NH_3N, TKN, and organic loading were 200 mg/L, 16.9 mg/L, and 27 mg/L, and 8,686 g BOD/m^3/day. The last model shows that the effluent BOD_5 is 2.272 mg/L and the influent TSS, NH_3N, TKN, organic loading, and flow rate were 198 mg/L, 17.5 mg/L, 27.5 mg/L, 8,944 g BOD/m^3/day, and 43 mgd.

The width of the graph shows the effect of inputs on output values. The widest one is NH_3H, which means that it is the most influential input parameter. This finding indicates that there may not be sufficient DO in aeration basins to biodegrade dissolved organics when the influent NH_3N concentration is high ranging from 25 to 40 mg N/L, as discussed later in Figure 5.20. The colors of the plot show the relationships of the inputs to the output (predicted effluent BOD_5). The blue color indicates a positive relationship, and the red color implies a negative relationship. In these figures, both the flow rate and organic loading have a positive relationship with the output. It means that the effluent BOD_5 is higher when the flow rate and organic loading are high, which is similar to influent BOD_5, TKN, and NH_3N. However, unlike influent TSS and TP parameters, higher TSS and TP values decreased predicted effluent BOD_5 values. Less DO is needed because more organics are in the solid phase and readily biodegradable soluble organics are uptaken by phosphorus-accumulating organisms (PAOs) in the anaerobic zone.

Figure 5.19 is the collective force plot but selecting one input value. The result shows the average of the inputs and the predicted output. The means of inputs can interpret the prediction result that when the effluent BOD_5 is 2.803 mg/L, the influent BOD_5, NH_3N, and TP were 212.3, 17.5, and 6.415 mg/L, respectively. The average values of inputs can help determine the system performance and average influent loading rate of the plant. The operator can determine and correct the problems of the system. The most influential parameter to the effluent BOD_5 prediction was NH_3N. In this case, more oxygen should be supplied to aeration basins so that nitrifiers can oxidize NH_3N.

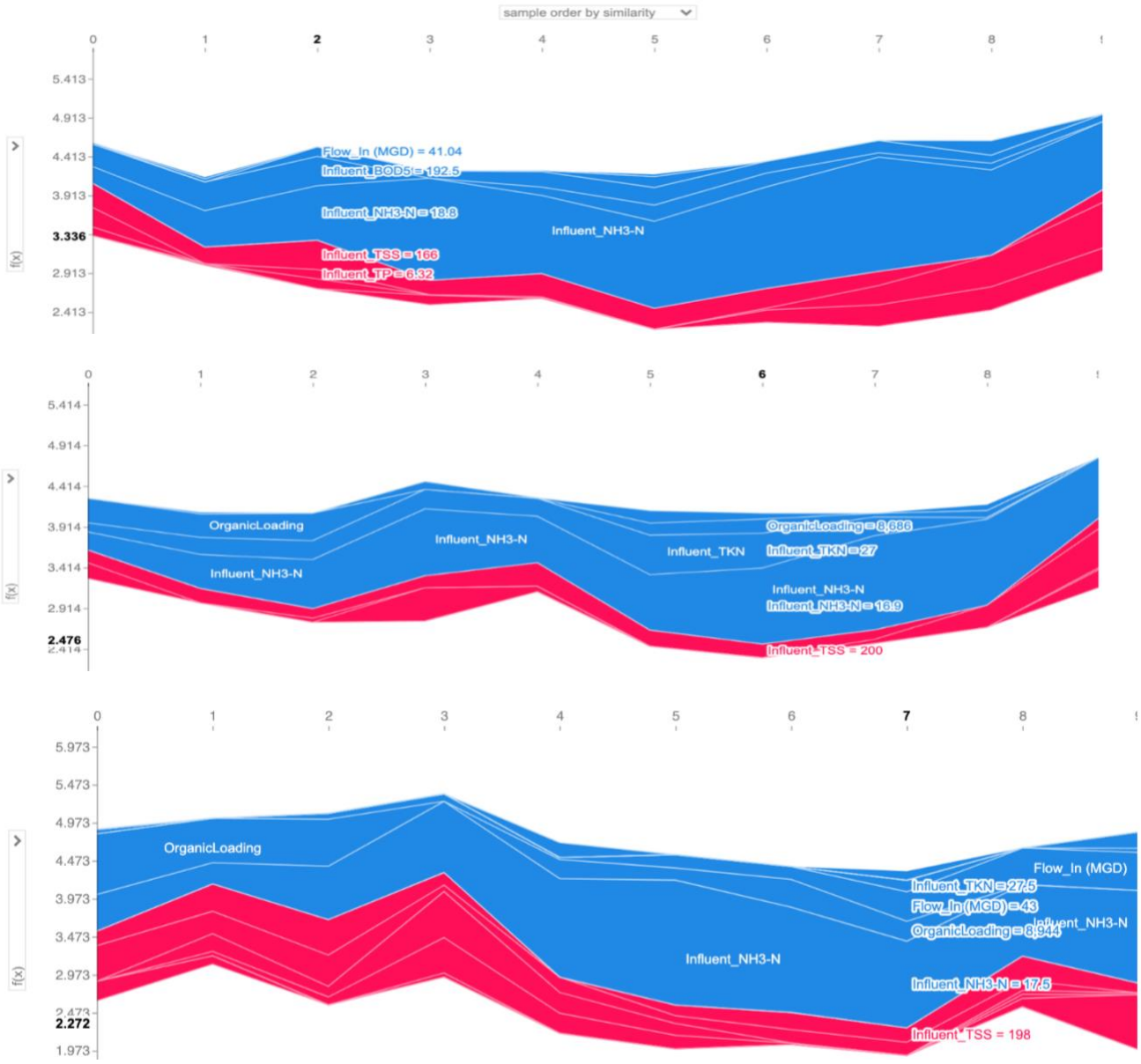


Figure 5.18 The inputs values in the collective force plot with different inputs.

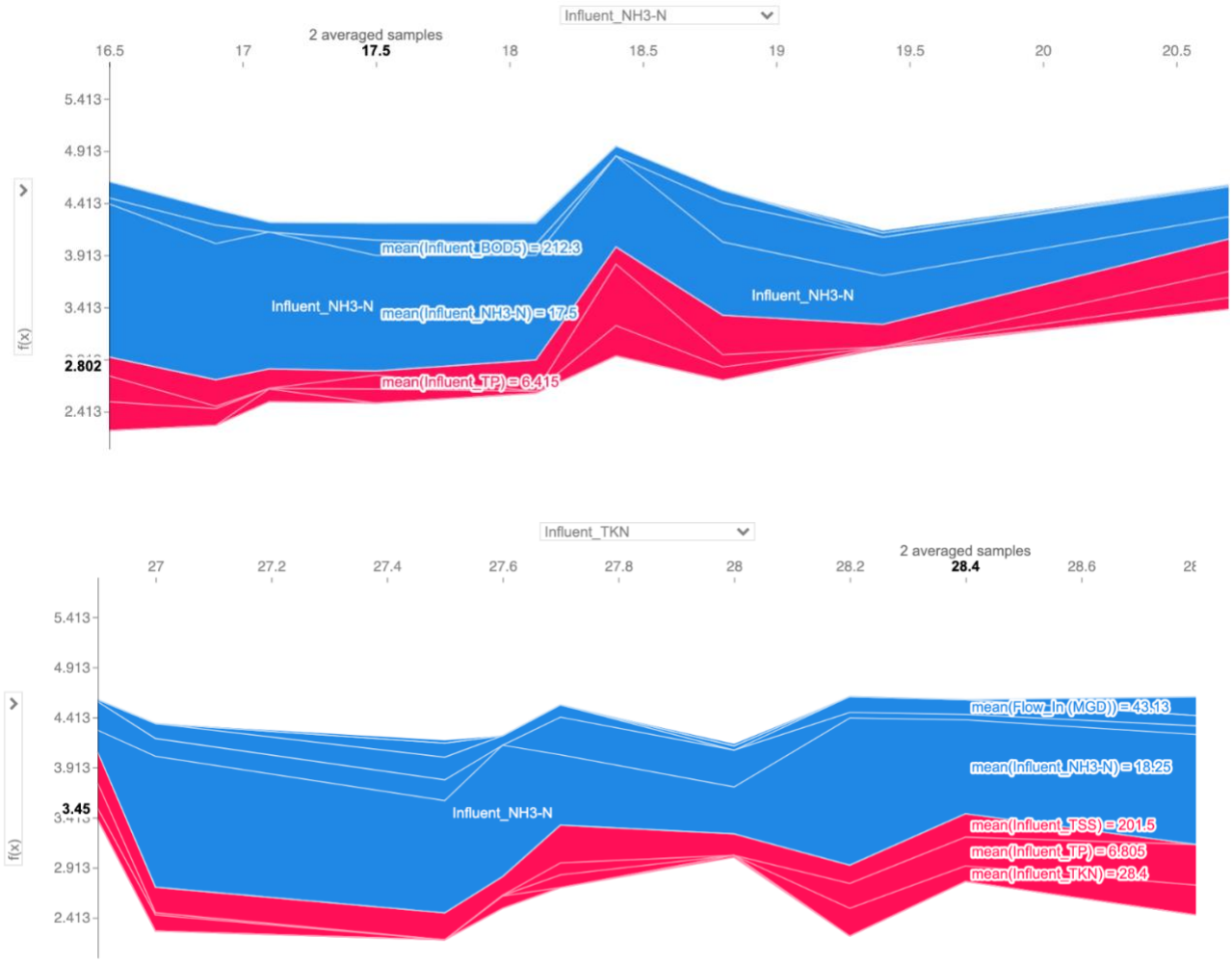


Figure 5.19 The mean of inputs values in the collective force plot by selecting one input value.

The SHAP function can create a SHAP dependence plot to show the effect of a single feature across the other features. In other words, the plot can show the effect between two inputs on the output. Figure 5.20 displays the SHAP dependence plot of influent NH_3N and BOD_5 to effluent BOD_5 prediction. The dependence plot is a scatter plot that shows the effect a single feature has on the prediction made by the model. The plot compares two chosen features and shows if these two features have an interaction effect. Each dot corresponds to one prediction. The x-axis is the influent NH_3N, while the y-axis is the predicted SHAP values of effluent BOD_5. The color represents influent BOD_5. Therefore, the plot can show if the influent NH_3N and influent BOD_5

positively or negatively impact the output. The figure shows that low influent NH_3N (between 15 to 25 mg/L) with low influent BOD_5 contributed to lower SHAP values, implying a lower effluent BOD_5 prediction. Moreover, the high influent NH_3N between 25 and 40 mg/L and high influent BOD_5 increased predicted effluent BOD_5 values.

```
shap.dependence_plot('Influent_NH3-N', shap_values[0], df_train, interaction_index='Influent_BOD5')
```

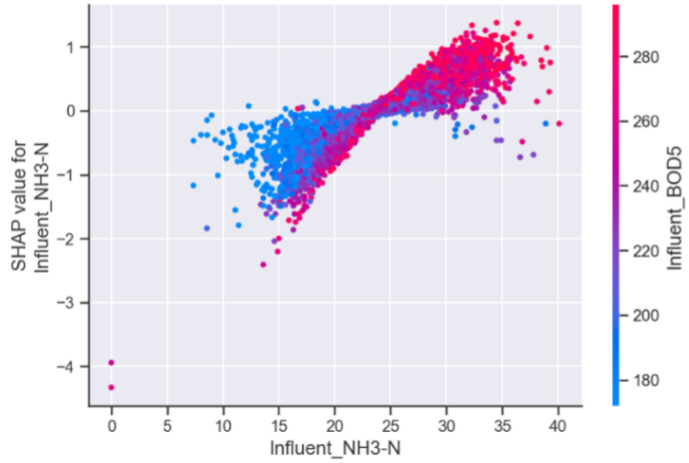


Figure 5.20 SHAP dependence plot of influent NH_3N to influent BOD_5.

Figure 5.21 shows the dependence plot of influent TKN and BOD_5 to effluent BOD_5 prediction. Low influent TKN values (between 20 to 37 mg/L) with low influent BOD_5 values resulted in lower effluent BOD_5 values. Conversely, high influent TKN values between 37 to 60 mg/L with high influent BOD_5 values predicted higher effluent BOD_5 values.

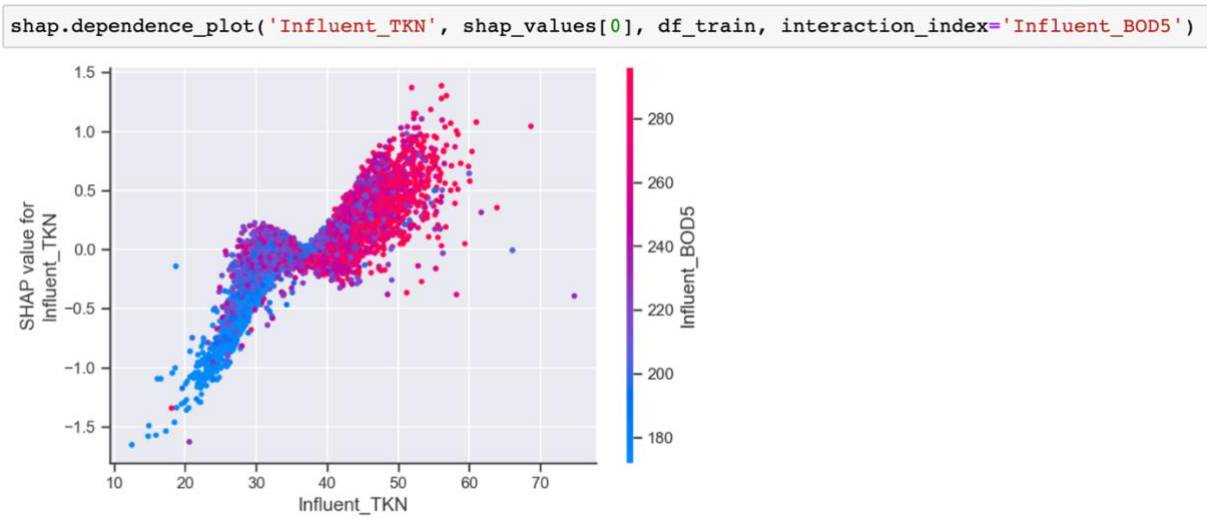


Figure 5.21 SHAP dependence plot of influent TKN to influent BOD_5.

Figure 5.22 shows the diagram of the real-time logistics for wastewater treatment operations (Krejcik, 2020). First, the data needs to be collected by physical devices from each system in the WWTP. Next, the data is described and visualized for the WWTP monitoring system. Then, the data is diagnosed and cleaned to be ready for a prediction model. After predicting the effluent water quality parameters, the data is applied to the SHAP function to explain the prediction results. Lastly, it is ready for real-time application in wastewater treatment facilities.

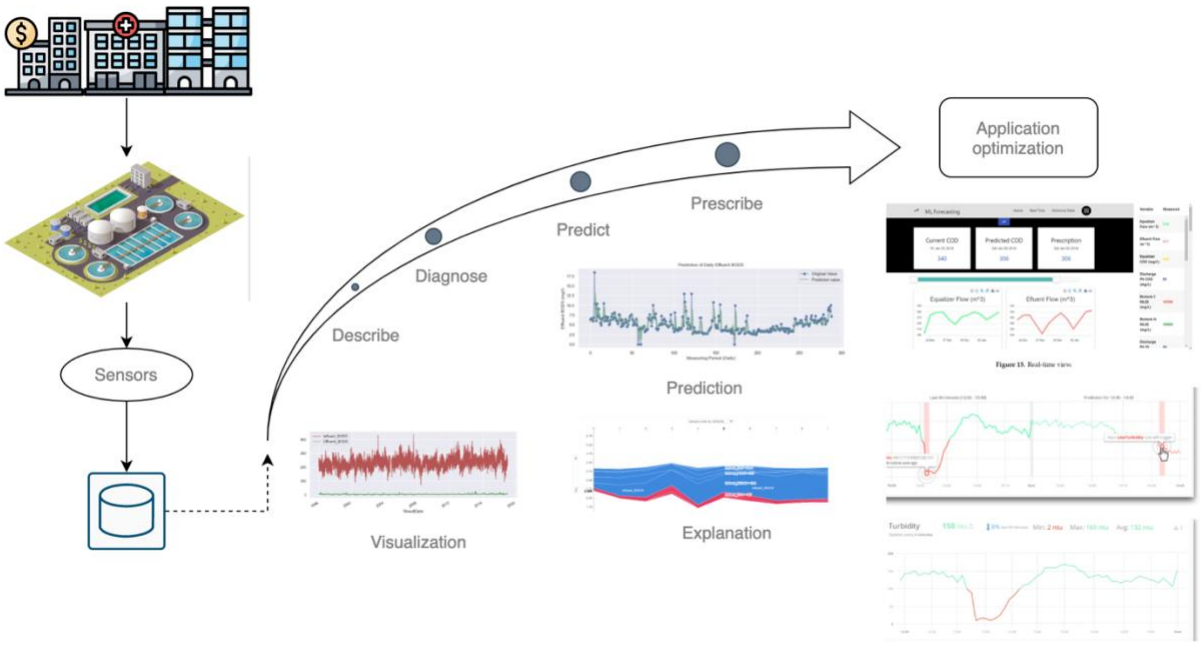


Figure 5.22 The diagram of the real-time logistics for wastewater treatment operation. (Modified from Krejcik, 2020)

Figure 5.23 shows the summary of the real-time logistics of the prediction models. First, the data is collected, called the data perception step. The data is collected from various devices such as water quality sensors, velocity sensors, etc. The second step is the data transmission by wired or wireless networks within the plant. Then, the data is stored in the database. The database is separated into different categories, such as the influent database, the preliminary data, the primary data, the effluent data, and the laboratory data to be ready to clean and apply in a model. The third step is data analysis. The data is analyzed and shown, called data visualization. Finally, a reliable dataset is created for the model prediction. Many model architectures can be applied in the last step. RNN architecture was used in this study. Lastly, real-time applications in WWTPs were implemented.

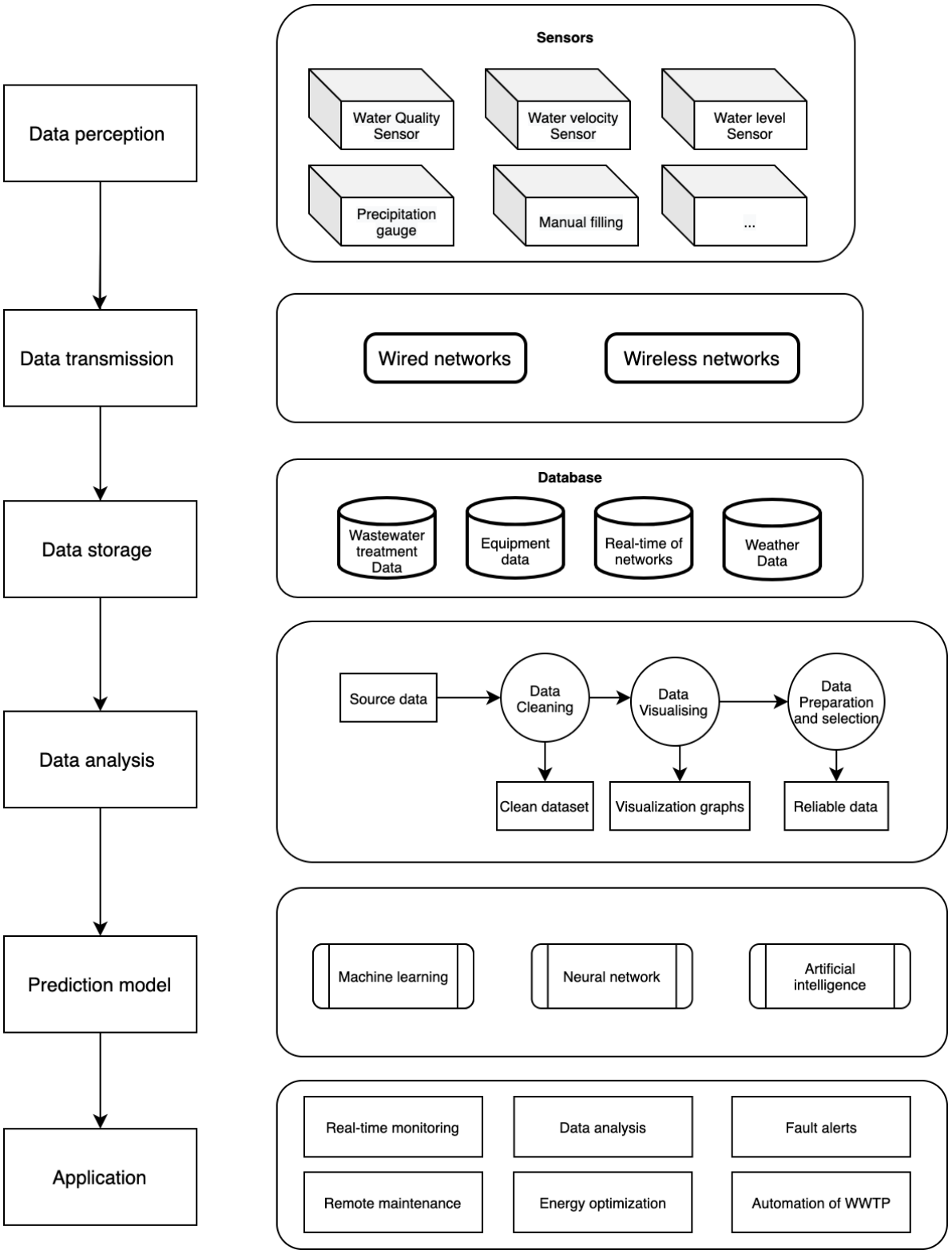


Figure 5.23 The real-time logistics for the prediction models in WWTPs.

5.5 Conclusions and Recommendations

Deep neural networks (DNNs) with real-time prediction modeling can be an intelligent tool for early warning of potential operational errors and subsequent effluent quality permit violations. Therefore, real-time logistics using RNNs are proposed in this study to help a WWTP's operator monitor, control, operate proactively, make decisions, and optimize WWTPs.

There are five important steps for the real-time logistics plan. The first step is data collection from each system in the plant. The second step is data visualization for monitoring the facility. The third step is data analysis by analyzing and cleaning the data for the next step, a prediction model. The RNN algorithms are applied for prediction. Next, the data is applied to the SHAP function, an explainable artificial intelligence algorithm, to explain the prediction results. Finally, it is used for real-time application in wastewater treatment facilities.

From this study, it can be concluded that the RNN model with Python can be implemented to a WWTP for operation and process optimization. The critical step was proven to be the data selection. The selection of inputs and appropriate datasets improved model performance significantly. In this study, the data was cleaned and analyzed so that the proper dataset is created and found the hidden meaning of the prediction, as proven in the previous chapter because the model was the same, but the Explainable AI function was added.

From the analysis of Nine Springs WWTP operational data, the Explainable AI analysis found that higher influent NH_3N values lead to higher effluent BOD_5 and higher influent TSS and TP values resulted in lower effluent BOD_5. The effluent BOD_5 is thought to be affected by the DO level in aeration basins. When the influent NH_3N concentration is high, more oxygen should be supplied to aeration basins. When the influent TSS concentration is high, more organics are associated with solids, leading to lower oxygen demand and subsequent lower effluent organics concentration

(BOD_5). More dissolved organics are uptaken by phosphorus-accumulating microorganisms when the influent TP is high, leading to lower effluent BOD_5. It is believed that if DO is well controlled, the effects of these parameters on effluent BOD_5 will be less noticeable. Thus, it is recommended that the Nine Springs WWTP considers further investigation on more robust control of DO based on NH_3N, TSS, and TP in real-time.

Applying an intelligent program to WWTP operations using a modern SCADA with Python will help support real-time control of a WWTP. The WWTP needs to develop the data collection scheme with the main point of energy and chemical controls. By implementing a SCADA system with Python, a WWTP can increase operational efficiency through monitoring real-time data and predicting operational parameters. The advanced control system can increase situational responsiveness, error detection, and efficiency whereas diminishing the possibility of errors. The modern SCADA system is not just monitoring and visualization with alarms rolling in. It is about optimizing operations for proactive identify problems and resulting in a beforehand resolution.

5.6 Future Research

As found from the Explainable AI analysis, future research should focus on developing real-time control of DO in aeration basins and subsequent energy optimization in WWTPs due to significant operational cost savings. Minimizing energy consumption is one of the challenges in the wastewater treatment sector. The real-time DO control should be the main focus in future work. Especially in the activated sludge process, a large amount of air is added to aeration basins to provide oxygen for enhancing removal efficiencies of BOD_5, ammonia, nitrate/nitrite, total phosphorus, and TSS. Blowers or low-pressure compressors typically inject a high volume of air through an aeration system. Blowers used in supplying air to aeration basins account for 45% to 60% of the total energy consumption in WWTPs (Wei, 2013). Therefore, an advanced

methodology of aeration optimization is crucial. Implementing a real-time DO control with SCADA and Python programs using RNNs algorithms and big data analysis would benefit the WWTP by closely monitoring DO in the process, achieving the effluent standard, and minimizing energy consumption in the activated sludge system.

6. PREDICTION OF SLUDGE VOLUME INDEX (SVI) IN A WASTEWATER TREATMENT PLANT USING ARTIFICIAL INTELLIGENCE

6.1 Abstract

Sludge Volume Index (SVI), defined as 'the volume (in mL) occupied by 1 gram of activated sludge after settling the aerated liquid for 30 minutes' (Wikipedia, 2020), is one of the most important operational parameters in an activated sludge process. It is difficult to predict SVI because of the nonlinearity of data and variability operation conditions. With complex time-series data from wastewater treatment plants (WWTPs), the Recurrent Neural Network (RNN) was applied to develop prediction models for SVI in a wastewater treatment process. RNN architecture has been proven that it can efficiently handle time-series and non-uniformity data. Data was collected from a WWTP, and the data were analyzed and cleaned using Python program data analytics approaches. An RNN model predicted SVI accurately after training with historical big data collected at a WWTP. Furthermore, the Explainable Artificial Intelligence (AI) analysis was able to determine which input parameters affected higher SVI most. The prediction of SVI will benefit WWTPs to establish corrective measures to maintaining stable SVI. The SVI prediction model and explainable artificial intelligence method will benefit the wastewater treatment sector to improve operational performance, system management, and process reliability.

6.2 Introduction

The activated sludge process has been used worldwide for wastewater treatment. One of the critical steps is to separate activated sludge from liquid before discharge. Thus, SVI (mL/g), an indicator

of solid separation, is an important operational parameter. Unfortunately, there is no deterministic model to predict SVI accurately due to the unexplainable behavior of microorganisms causing the sludge bulking (settling) problem. A common problem in the activated sludge system is poor solid separation at the secondary clarification stage. Excess growth of filamentous organisms makes activated solids difficult to settle in secondary clarifiers, leading to a potential violation of the total solids regulatory limit. The recurrent neural network (RNN) model was used to predict SVI from big data generated at the Nine Springs WWTP, Madison, Wisconsin. The model will aid in predicting potential settling issues and providing possible reasons for higher SVI prediction. This model will significantly enhance the activated sludge system performance in WWTPs.

6.2.1 Background

Control of the activated sludge process is difficult for many WWTPs due to the complexity of the biological and chemical reactions and variations in the influent water quality and flow rate (Phuc et al., 2018). The activated sludge process control can be improved by evaluating the causes of higher SVI and taking preventive measures. Several variables can impact the settleability of sludge in the clarifiers, such as filamentous bacteria, rain events, and water temperature. Filamentous bacteria such as *S. natans* and *M. parvicella* impact sludge settleability because these bacteria create more buoyant flocs (Yousuf, 2013). Moreover, a balance between floc- and filamentous-forming bacteria is required. There are still many causes leading to problems in the activated sludge process. Therefore, sludge settling problems occurring in the activated sludge process can be avoided by taking the right action through the real-time monitoring system, Explainable AI algorithms, and proactive measures.

In recent years, predictive modeling approaches have been increasingly applied in many industries. Modeling with RNNs is a current trend in deep learning neural networks algorithms. The

advantage of RNN algorithms is the capability to handle sequential data with a variable dataset. SVI is one of the most important parameters that monitors the activated sludge settling performance in WWTPs. RNN models can be used to predict SVI in the activated sludge system with a validated dataset and then applied to an explainable function to interpret the result, allowing operators to take preventive measures before the sludge settling issue arises during the operation of the activated sludge process.

6.2.2 Sludge Volume Index

Operators use SVI to determine and compare mixed liquor settleability (Bye & Dold, 1998). It mathematically relates settled sludge volume in the settleometer to mixed liquor suspended solids (MLSS) concentration. SVI relates sludge volume in milliliters to MLSS concentration in grams per liter as follows (Tesh, 2016):

$$SVI\ (mL/g) = \frac{Settled\ Sludge\ Volume\ SSV_{30}\ (mL/L)}{MLSS\ (mg/L)} \times\ 1{,}000\ mg/g \qquad 6.1$$

where SSV_{30} (in units of milliliters per liter) is the volume of sludge that settles in a graduated cylinder of mixed liquor in 30 minutes and mixed liquor suspended solids (MLSS) (mg/L) is the MLSS concentration in aeration basins. The common range for an SVI at a conventional activated sludge system is between 50 and 150 (Finnegan, 2020). Optimum SVI must be determined for each WWTP experimentally. SVI is an excellent indicator of the settling characteristics of the sludge. However, SVI varies with the characteristics and concentration of the mixed-liquor solids. Thus, observed values at a given WWTP should not be compared with those reported for other plants or in the literature. Typical SVI values for a good settling sludge with 1,500 to 3,500 mg/L of mixed liquor concentrations range from 80 to 120 (Finnegan, 2020). Filamentous bulking

increases SVI even if the MLSS concentration is the same. Therefore, SVI is a good indicator of filamentous bulking.

6.2.3 Activated Sludge Process

The activated sludge process is a biological wastewater treatment process where microorganisms biodegrade organics present in wastewater as a carbon source. The settleability of the activated sludge depends on the size, density, and shape of the flocs and the competency of the secondary clarifier. Settleability can be affected by the extent of the filamentous bacteria population. These bacteria can form strings as they grow rather than forming flocs. Excess growth of these filamentous organisms can cause a bulking condition, resulting in poor settling and taking up more sludge blanket volume in the secondary clarifier. This condition may be triggered by several factors, such as inadequate DO and nutrient imbalance, leading to solids loss in the clarifier effluent due to poor solid separation (Jenkins et al., 2003). Therefore, the control of sludge bulking is crucial in the activated sludge process.

6.2.4 Filamentous Bulking

Filamentous bulking is the number one cause of effluent non-compliance in the United States (U.S.) Filamentous bulking and foaming are serious issues in an activated sludge operation, affecting most WWTPs (Jenkins et al., 2003). A bulking sludge settles slowly and doesn't settle compactly, causing subsequent solids overflow at the secondary clarifier. An operational target SVI often used for operation is $<$ 150 mL/g, although each WWTP has unique SVI values for safe operation, varying from $<$ 100 mL/g to $>$ 300 mL/g, depending on the hydraulic considerations and the capacity and performance of the secondary clarifier. Or example, a bulking sludge may be acceptable if the secondary clarifier is sufficiently large.

6.3 Materials and Methods

Data Collection

The dataset from 1997 to 2020 was obtained from the Nine Springs WWTP, Madison, Wisconsin, in the U.S. Table 6.1 summarizes the data from the Nine Springs WWTP. After removing missing rows, there are 30 columns of parameters, including flow rate, influent parameters, effluent parameters, SVI, sludge age, and 8,642 rows of data.

Table 6.1 Nine Springs Wastewater Treatment Dataset

```
<class 'pandas.core.frame.DataFrame'>
Index: 8642 entries, 1996-07-29 to 2020-03-05
Data columns (total 30 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Flow_In (MGD)        8642 non-null   float64
 1   Influent_BOD5        8642 non-null   float64
 2   Influent_CBOD5       8642 non-null   float64
 3   Influent_TSS         8642 non-null   float64
 4   Influent_TKN         8642 non-null   float64
 5   Influent_NH3-N       8642 non-null   float64
 6   Influent_TP          8642 non-null   float64
 7   Influent_VSS         8642 non-null   float64
 8   Influent_pH          8642 non-null   float64
 9   Flow_Eff (MGD)       8642 non-null   float64
 10  Effluent_BOD5        8642 non-null   float64
 11  Effluent_TSS         8642 non-null   float64
 12  Effluent_TKN         8642 non-null   float64
 13  Effluent_NH3-N       8642 non-null   float64
 14  Effluent_TP          8642 non-null   float64
 15  Effluent_VSS         8642 non-null   float64
 16  Effluent_Temp        7906 non-null   float64
 17  Effluent_DO          7906 non-null   float64
 18  SVI_Plant1           8642 non-null   float64
 19  SVI_Plant2           8642 non-null   float64
 20  SVI_Plant3           8642 non-null   float64
 21  SVI_Plant4           8642 non-null   float64
 22  SludgeAge_Plant1     8642 non-null   float64
 23  SludgeAge_Plant2     8642 non-null   float64
 24  SludgeAge_Plant3     8642 non-null   float64
 25  SludgeAge_Plant4     8641 non-null   float64
 26  SludgeAge_Plant1_AO  8642 non-null   float64
 27  SludgeAge_Plant2_AO  8642 non-null   float64
 28  SludgeAge_Plant3_AO  8641 non-null   float64
 29  SludgeAge_Plant4_AO  8641 non-null   float64
```

The next step is to select the inputs of the dataset. In this study, the inputs were the flow rate, influent BOD_5, TSS, TKN, NH_3N, TP, and organic loading (flow rate × influent BOD_5) and the output was SVI. Table 6.2 shows the dataset from 1996 to 2020.

Table 6.2 Dataset from 1996 to 2020.

Date	Flow_In (MGD)	Influent_BOD5	Influent_TSS	Influent_TKN	Influent_NH3-N	Influent_TP	OrganicLoading	SVI_Plant1
1996-07-29	41.3600	210.0	200.0	27.0	16.9	6.13	8685.6000	95.12200
1996-07-30	41.6200	220.5	215.0	28.2	16.5	6.56	9177.2100	91.37060
1996-07-31	40.7100	216.5	245.0	28.8	17.5	6.98	8813.7150	92.80740
1996-08-01	41.5300	229.0	216.0	28.4	18.4	7.33	9510.3700	79.21540
1996-08-02	41.0400	192.5	166.0	27.7	18.8	6.32	7900.2000	85.35130
...	...	...	...	...	...	...	...	...
2020-03-02	42.1198	229.0	247.0	42.8	28.1	5.38	9645.4342	107.78805
2020-03-03	42.1532	219.0	75.6	41.3	29.4	4.30	9231.5508	112.00470
2020-03-04	42.3383	219.0	197.0	47.2	31.1	5.36	9272.0877	116.22135
2020-03-05	42.6054	219.0	197.0	47.2	32.7	5.36	9330.5826	120.43800
2020-03-05	42.6054	219.0	197.0	47.2	32.7	5.36	9330.5826	120.43800

Data visualization was applied using the Python program. Figure 6.1 displays SVI data from 1996 to 2020. Then, the box plot analysis was used to analyze the yearly data, as shown in Figure 6.2. The plot shows that in 2000, there are many errors in SVI data. Therefore, the dataset from 2001 to 2020 was used. Figure 6.3 shows SVI data from 2001 to 2020. It appears that the maximum value of SVI data decreases from 1,000 mL/g to 200 mL/g. SVI data from 2001 to 2020 are plotted in Figure 6.4. Therefore, the dataset was more stable and had fewer errors. In this box plot, it can be seen that 2001, 2003, 2008, and 2009 have a wide range of SVI values, implying that the process was unstable.

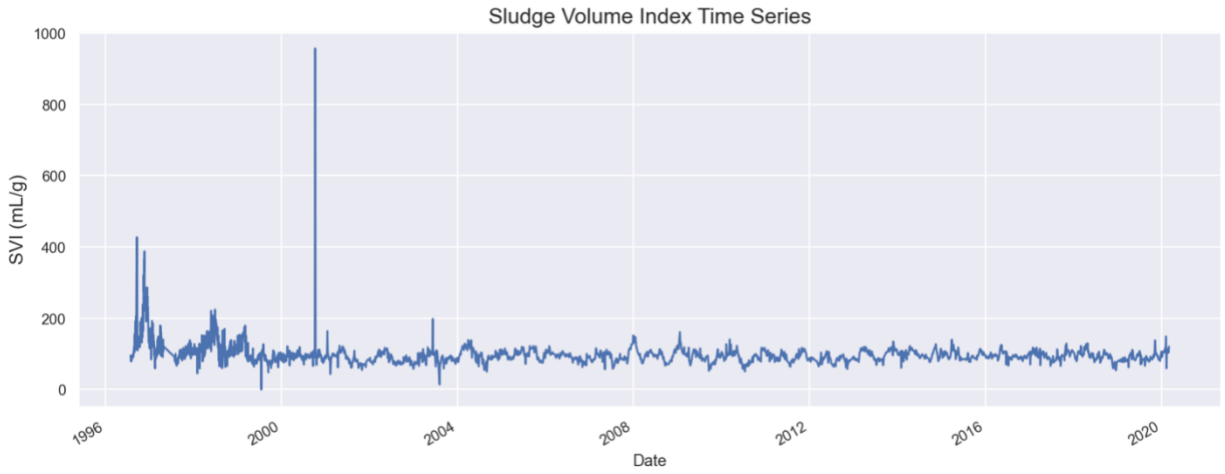


Figure 6.1 Sludge Volume Index data from 1996 to 2020

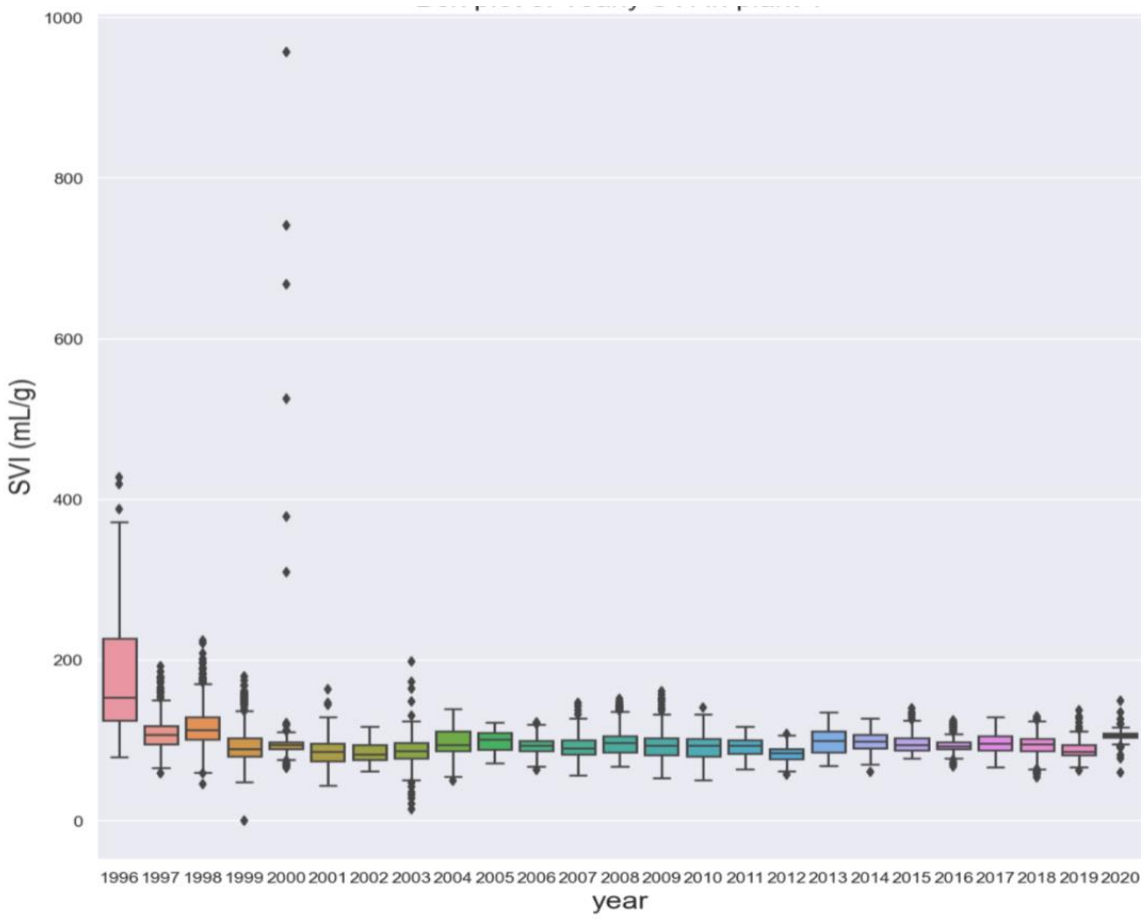


Figure 6.2 Box plot of yearly SVI from 1996 to 2020.

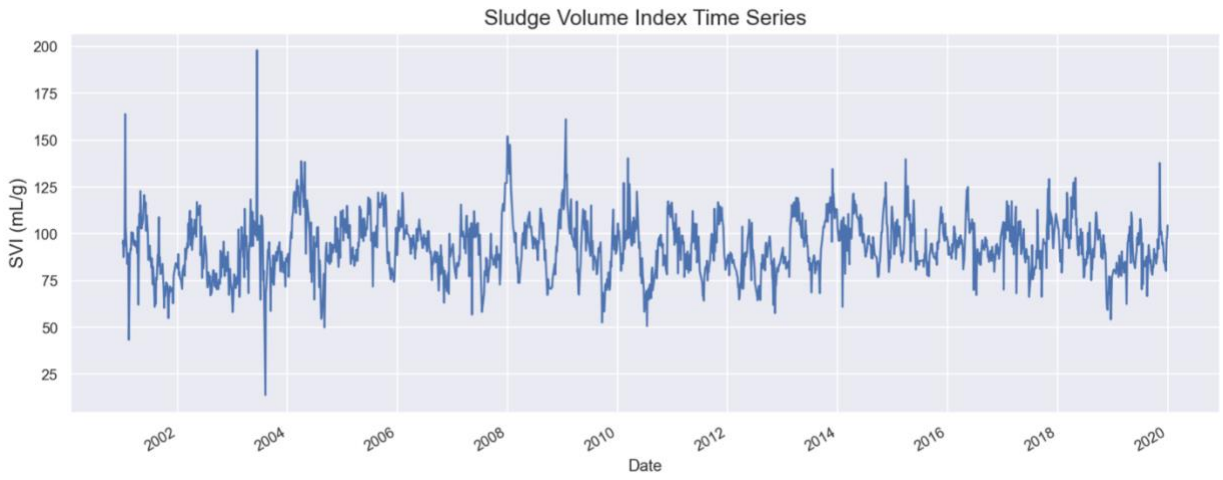


Figure 6.3 Sludge Volume Index data from 2001 to 2020.

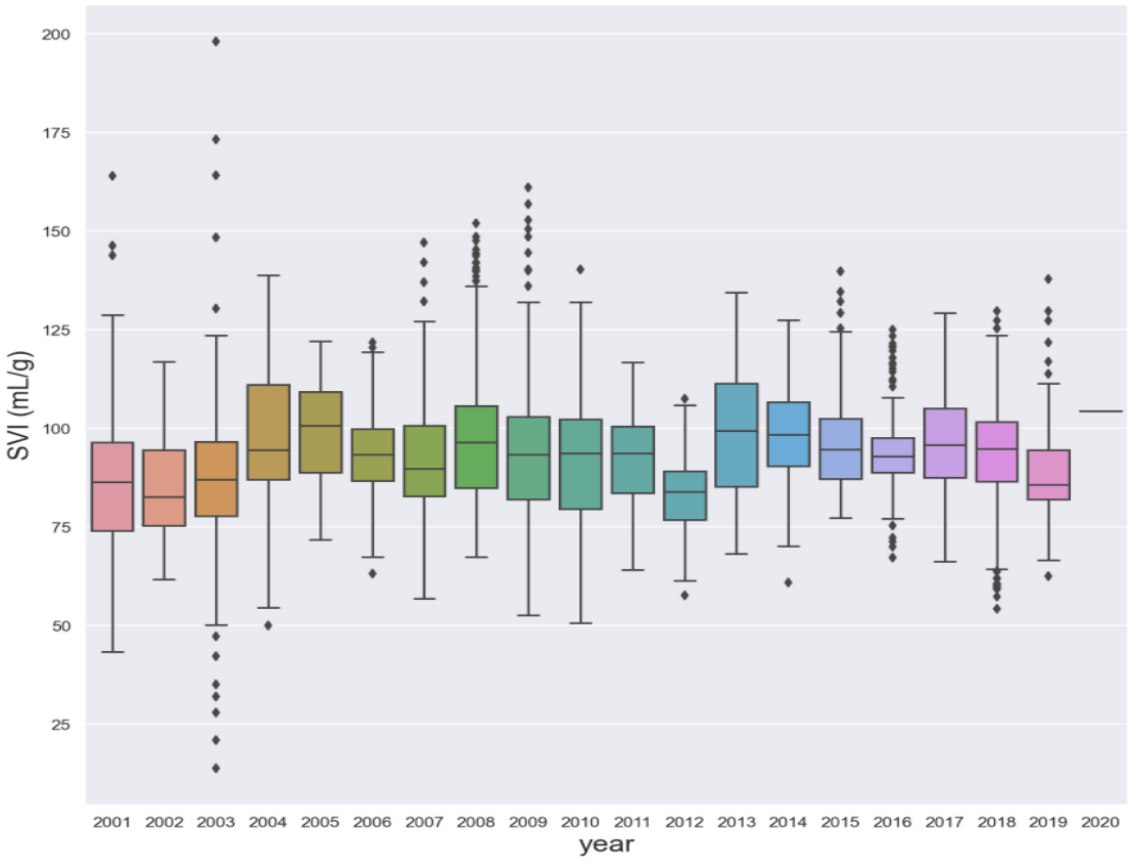


Figure 6.4 Box plot of yearly SVI from 2001 to 2020.

Therefore, the third dataset was selected from 2010 to 2020, which shows more stable data with the appropriate range of SVI of 50 to 150 mL/g. Figure 6.5 shows the SVI time-series data from 2010 to 2020. Figure 6.6 shows a box plot of yearly SVI from 2010 to 2020.

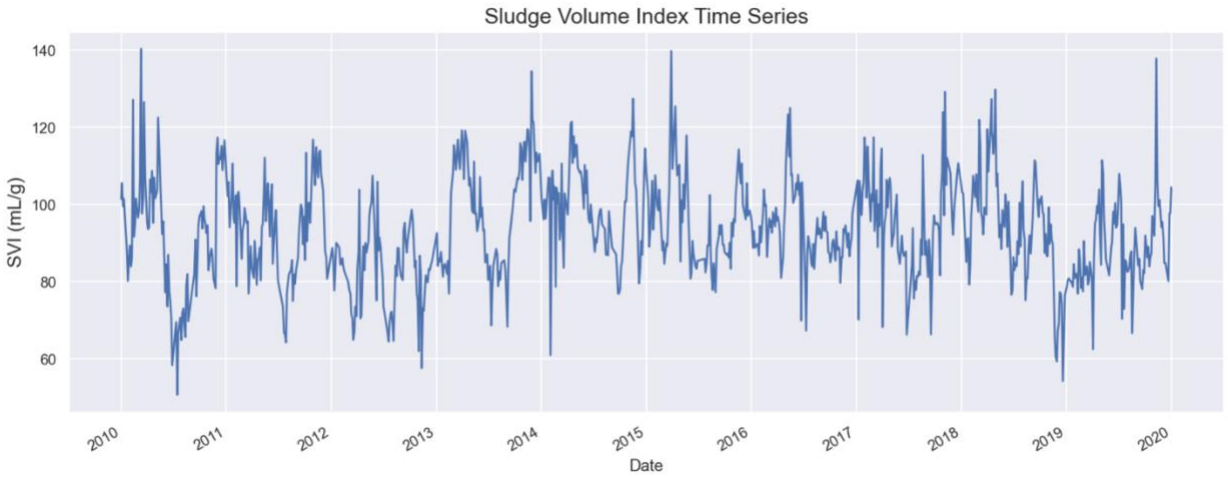


Figure 6.5 Sludge Volume Index data from 2010 to 2020.

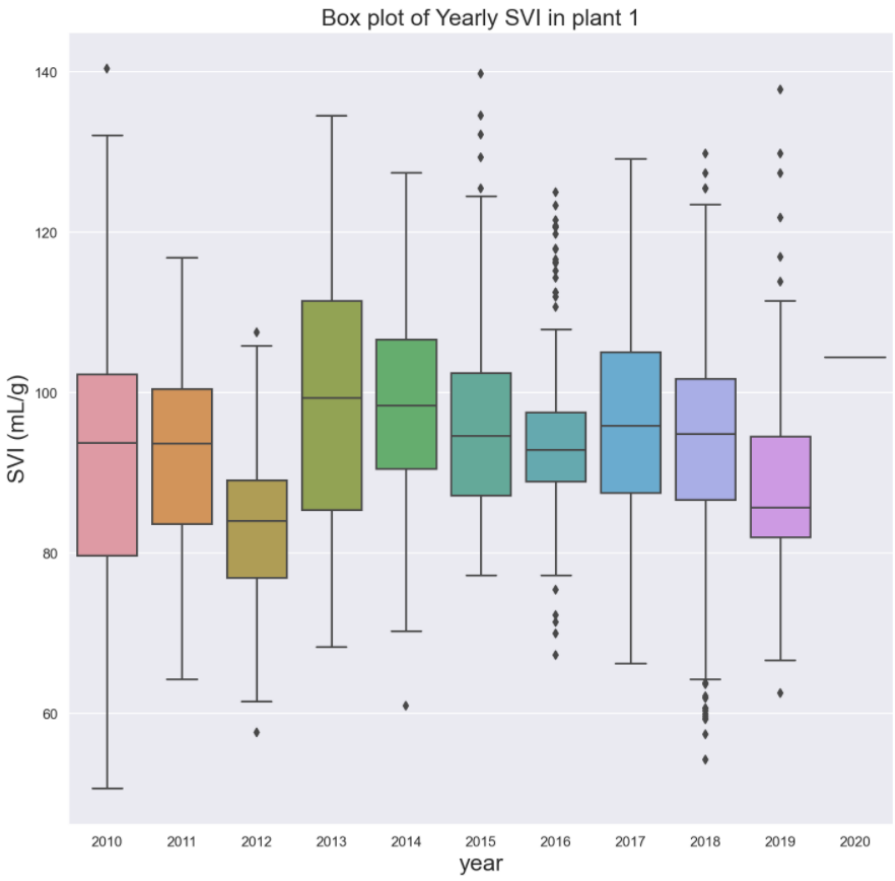


Figure 6.6 Box plot of yearly SVI from 2010 to 2020.

Recurrent Neural Networks (RNN) Model

The RNN model was selected, and the inputs and output values were normalized (from 0 to 1) as shown in Table 6.3. Then, the dataset was separated into training (80% of the dataset) and testing (20% of the dataset) sets and put in the model. Figure 6.7 shows the RNN model in the Python program.

Table 6.3 Normalization inputs and output values of the models.

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var6(t-1)	\
1	0.631085	0.192411	0.003406	0.563376	0.563376	0.369318	
2	0.621351	0.078167	0.014519	0.423265	0.423265	0.306818	
3	0.611617	0.005066	0.004602	0.364614	0.364614	0.332386	
4	0.601882	0.076612	0.013616	0.423265	0.423265	0.318182	
5	0.589009	0.066292	0.028674	0.387423	0.387423	0.315341	

	var7(t-1)	var8(t-1)	var9(t-1)	var1(t)
1	0.734654	0.783042	0.591954	0.621351
2	0.645545	0.783042	0.568966	0.611617
3	0.681188	0.783042	0.571839	0.601882
4	0.546535	0.750624	0.540230	0.589009
5	0.534653	0.678304	0.581897	0.576135

```
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
```

```
(1472, 1, 9) (1472,) (368, 1, 9) (368,)
```

```
model = Sequential()
model.add(SimpleRNN(50, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
history = model.fit(train_X, train_y, epochs=100, batch_size=10,
                    validation_data=(test_X, test_y), verbose=2, shuffle=False)
```

```
Epoch 1/100
148/148 - 0s - loss: 0.1211 - val_loss: 0.0785
Epoch 2/100
148/148 - 0s - loss: 0.0726 - val_loss: 0.0490
Epoch 3/100
148/148 - 0s - loss: 0.0576 - val_loss: 0.0323
Epoch 4/100
148/148 - 0s - loss: 0.0341 - val_loss: 0.0382
Epoch 5/100
148/148 - 0s - loss: 0.0288 - val_loss: 0.0645
Epoch 6/100
148/148 - 0s - loss: 0.0310 - val_loss: 0.0273
Epoch 7/100
148/148 - 0s - loss: 0.0272 - val_loss: 0.0248
Epoch 8/100
148/148 - 0s - loss: 0.0249 - val_loss: 0.0411
Epoch 9/100
148/148 - 0s - loss: 0.0256 - val_loss: 0.0500
Epoch 10/100
148/148 - 0s - loss: 0.0268 - val_loss: 0.0409
```

Figure 6.7 The Recurrent Neural Networks model prediction in Python.

SHapley Explanation

The last step is to apply the SHapley function. Table 6.4 shows the general statistics for the dataset: the number of datasets, the average, standard deviation, minimum and maximum values, and average percentile of data. Then, the correlation coefficient was applied in Figure 6.8 to determine the correlated number between each parameter.

Table 6.4 General statistics for the dataset from 2010 to 2020.

	SVI_Plant1	OrganicLoading	Flow_In (MGD)	Influent_BOD5	Influent_TSS	Influent_TKN	Influent_NH3-N	Influent_TP
count	3673.000000	3673.000000	3673.000000	3673.000000	3673.000000	3673.000000	3673.000000	3673.000000
mean	93.305396	9854.987777	41.141100	241.901062	222.853390	42.871606	27.190648	5.601525
std	12.627223	1487.355703	5.519575	39.023767	35.112722	5.531842	3.839846	0.767994
min	50.691200	5700.988800	28.613000	93.100000	110.000000	18.200000	0.000000	2.380000
25%	84.786800	8844.805300	37.731700	216.000000	201.000000	39.500000	24.800000	5.120000
50%	92.812914	9829.965300	40.095600	242.000000	220.000000	43.000000	27.500000	5.640000
75%	101.663100	10775.050000	43.216500	267.000000	242.000000	46.300000	29.800000	6.110000
max	140.350900	17395.760400	112.943400	400.000000	485.000000	74.800000	40.100000	16.600000

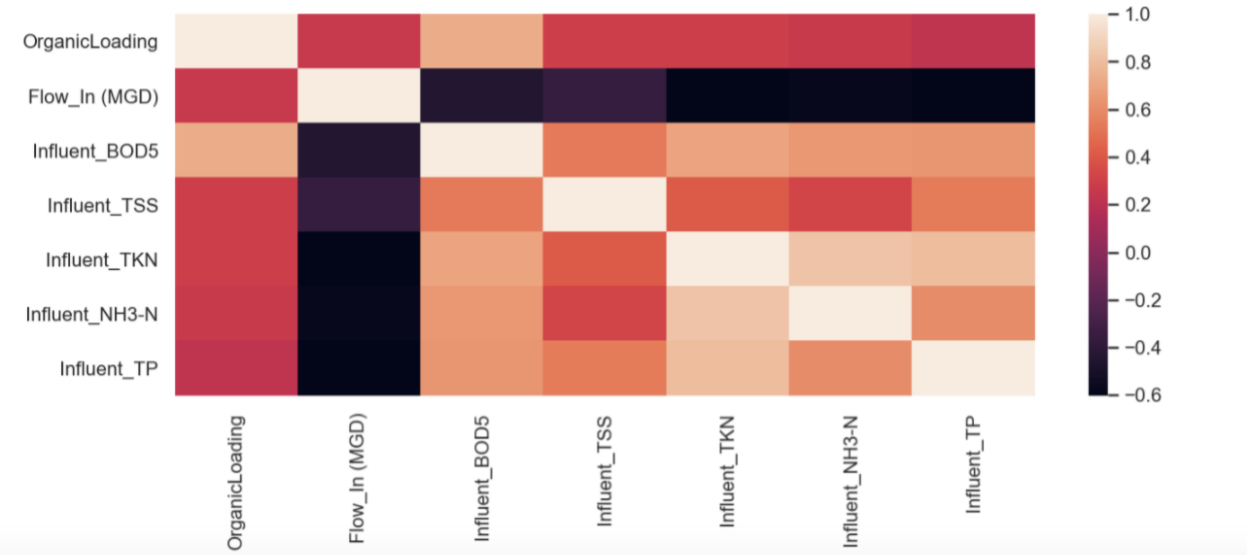


Figure 6.8 Correlation coefficient between each parameter.

If the value is positive, the parameters have a positive correlation. Conversely, the negative value (darker color) means negatively correlated to another parameter.

6.4 Results and Discussion

The first dataset is from 1996 to 2020, the original dataset from the Nine Springs WWTP. After data analysis, the second and third datasets were created. The second dataset is the dataset between 2001 and 2020 because the data in 2000 was found to have a significant error in the dataset. The

third dataset is the data from 2010 to 2020, in which the out-of-range (50 to 150 mL/g) SVI data were removed. The result shows that the second and third datasets are more suitable for applying in the model. Figures 6.9 to 6.11 show the normal distribution used to determine if the data distribution departs from the normal distribution. Kurtosis and Skewness tests were also calculated. Figure 6.9 shows that the Kurtosis of the first dataset was far from 0, implying that the distribution had heavier tails. Skewness measures the asymmetry of the distribution. If the Skewness is between -0.5 and 0.5, the data are symmetrical. If the skewness is between -1 and 0.5 or between 0.5 and 1, the data are moderately skewed. If the skewness is less than -1 or greater than 1, the data are highly skewed. Therefore, the first dataset was very highly skewed.

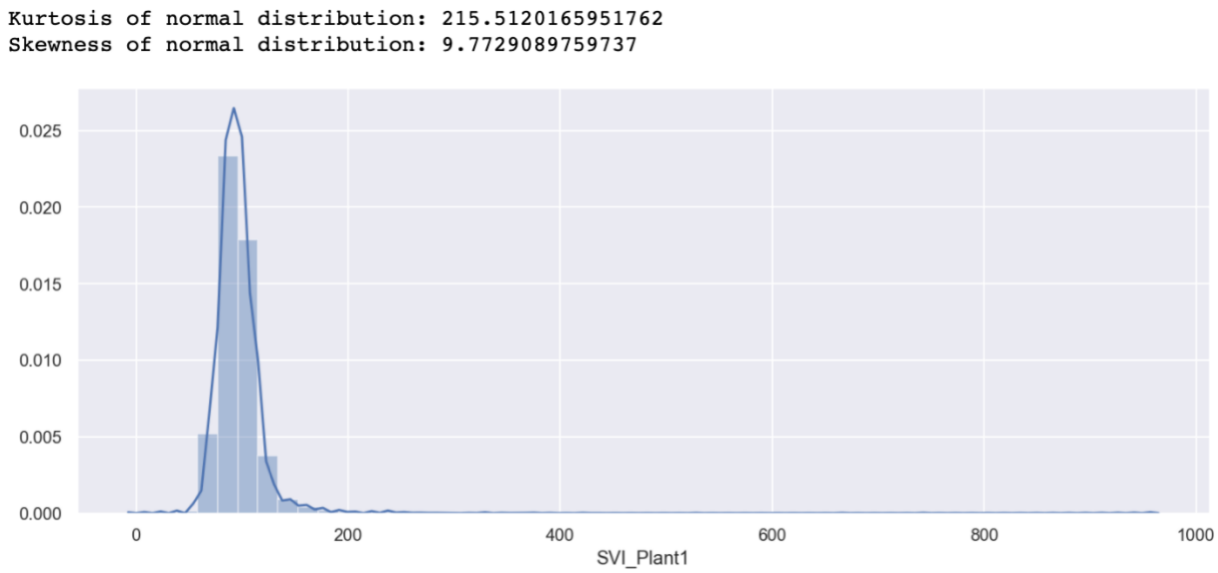


Figure 6.9 Normal distribution of the first dataset from 1996 to 2020.

Data visualization was performed to determine the appropriateness of the dataset. The data in 2000 appears to have a high error. Thus, the second dataset from 2001 to 2020 was used for modeling. Figure 6.10 shows the normal distribution and Kurtosis and Skewness values. The Kurtosis was closer to 0, which was decreased from 9.77 in the first dataset to 1.35 in the second dataset. The Skewness value displays that the second dataset is symmetrical.

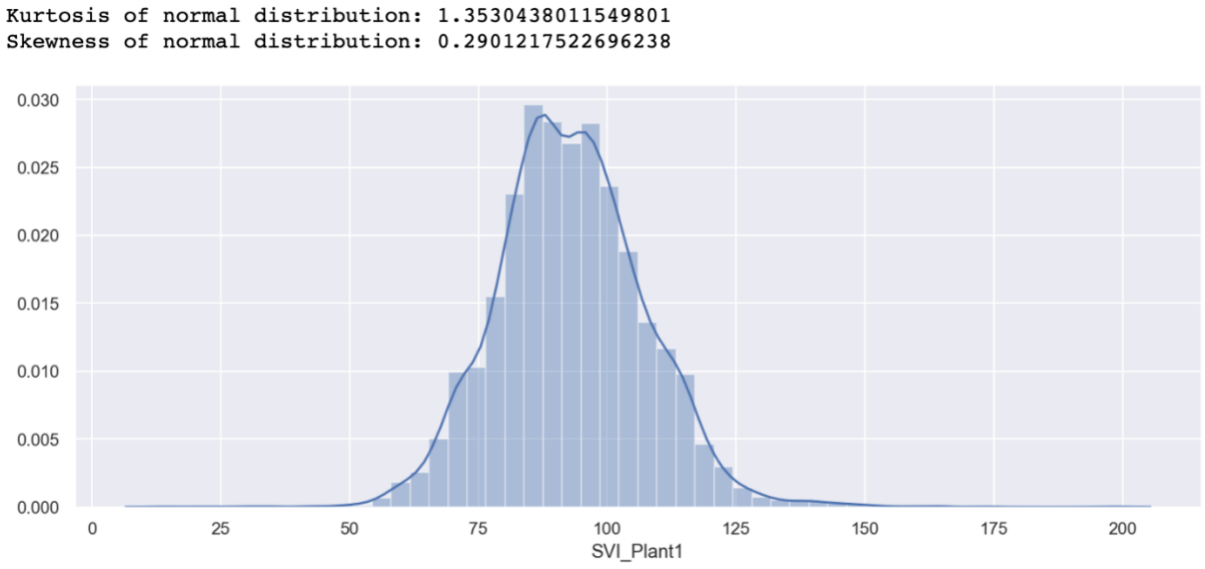


Figure 6.10 Normal distribution of the second dataset from 2001 to 2020.

Lastly, the third dataset's Kurtosis and Skewness values were calculated. Figure 6.11 shows that the dataset has the asymmetry of the distribution and symmetry of the dataset. The result has the Kurtosis of 0.03 and the Skewness of 0.12.

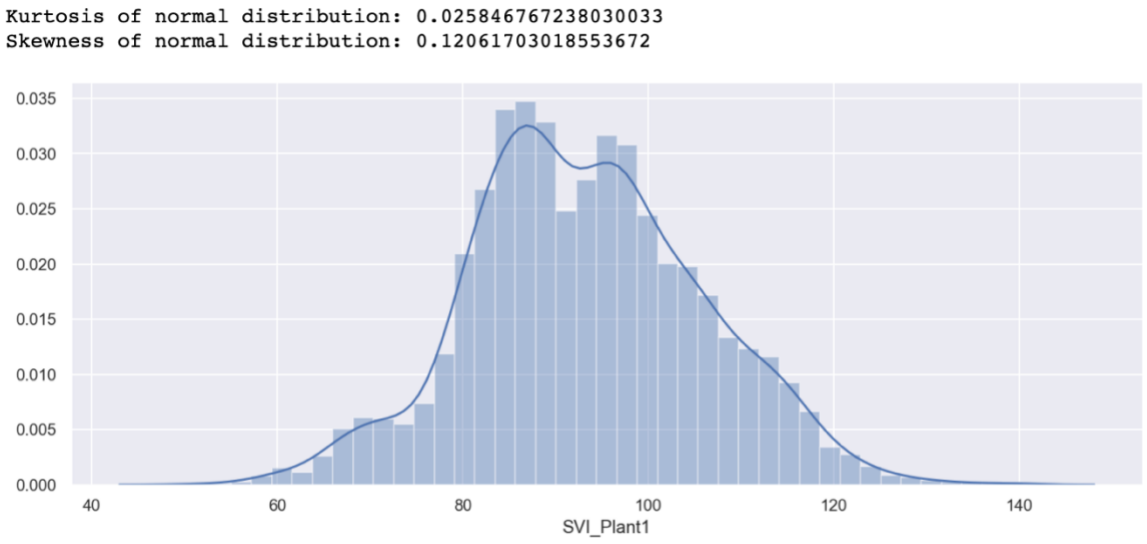


Figure 6.11 Normal distribution of the third dataset from 2010 to 2020.

The probability plots are shown in Figures 6.12 to 6.14. Figure 6.12 shows that the first dataset was far from the normal probability plot and had a high standard deviation (std) of 26.69, indicating that the dataset had significant errors. Figures 6.13 and 6.14 show an excellent inline of the probability plot for the second and third datasets. The standard deviations decreased to 14.36 and 12.63, respectively.

	count	mean	std	min	25%	50%	75%	max
SVI_Plant1	8642.0000	96.8289	26.6898	0.0000	84.5964	94.0125	104.1607	956.9378

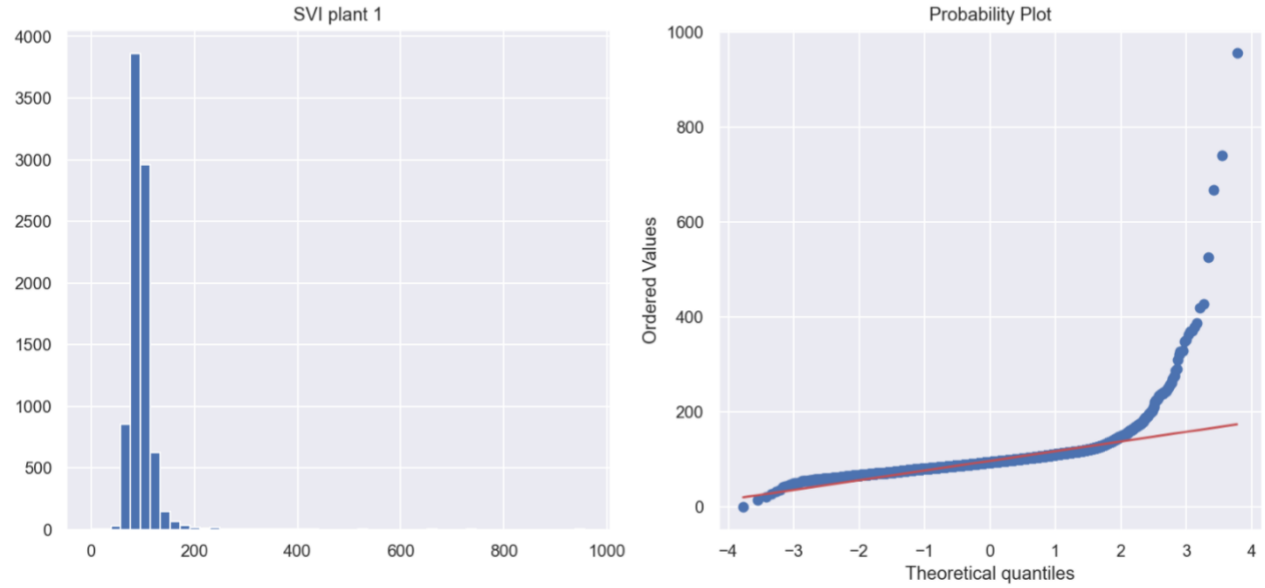


Figure 6.12 Normal probability plot of the first dataset.

	count	mean	std	min	25%	50%	75%	max
SVI_Plant1	6960.0000	92.9577	14.3596	13.8889	83.6604	92.5325	101.9006	198.0198

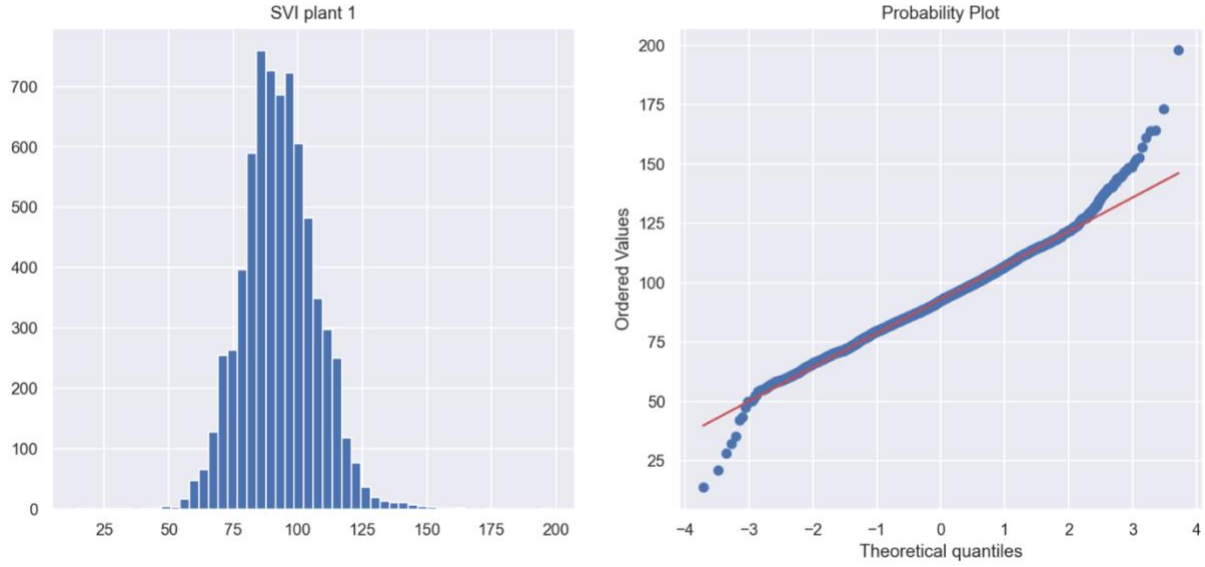


Figure 6.13 Normal probability plot of the second dataset.

	count	mean	std	min	25%	50%	75%	max
SVI_Plant1	3673.0000	93.3054	12.6272	50.6912	84.7868	92.8129	101.6631	140.3509

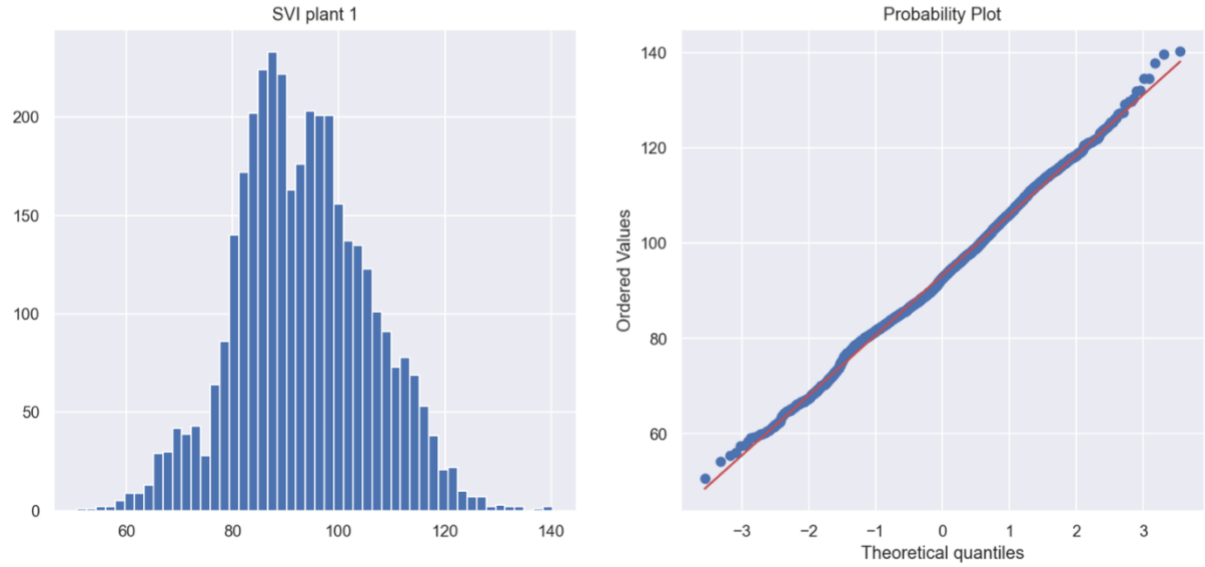


Figure 6.14 Normal probability plot of the third dataset.

The next step was the RNN model development. Figure 6.15 shows the train and validation error of the training and testing of the third dataset. It can be seen that the model performed well in the sample fit. The model fit can be ended when the train and test error is low and close to each other. The figure shows two lines of train results. The test errors were very close to each other.

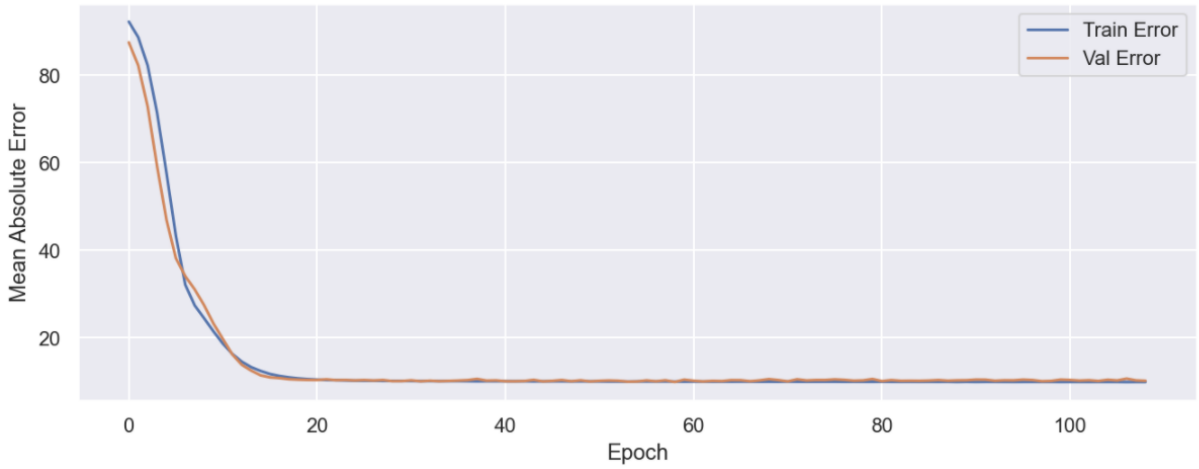


Figure 6.15 Mean Absolute Error between each epoch of the model.

Figures 6.16 to 6.18 show the prediction results of the SVI model. The blue line is the original SVI data, and the green line is predicted. RMSE and MAE are the most common metrics used to measure the accuracy of a model. RMSE is a quadratic scoring rule that measures the magnitude of errors, i.e., the square root of the average of squared differences between the prediction and original values. MAE is used to measure the average magnitude of the errors in the predictions. It calculates the absolute differences between the prediction and actual values over the test data where all individuals have equal weight. Therefore, both MAE and RMSE can be used to express average model prediction. The metrics range from 0 to ∞. Figure 6.16 shows that the prediction of the first set of data. The prediction has a low RMSE of 3.357 and MAE of 2.347.

The prediction of the second dataset after data analysis results in lower RMSE (3.076) and MAE (2.163) values in Figure 6.17. However, Figure 6.18 shows that even though the data are more

stable, the model's RMSE and MAE were similar or slightly higher than the previous model. Thus, the RNN model can perform well even if the data fluctuate. Data visualization and analysis can help determine the error in the dataset and the system's poor performance.

The next step is to interpret the prediction result of the models. Figures 6.19 to 6.21 show the explainable function applied to the three prediction models. Figure 6.19 (the first figure) shows that organic loading, BOD_5, and flow rate are the most impact input parameters to the SVI prediction, followed by TP, TKN, TSS, and NH_3N. Figure 6.19 (the middle figure) shows that when SVI is 114.6, organic loading and flow rate lowered the SVI prediction value, and TP, TSS, NH_3N, TKN, and BOD_5 increased the output. Lastly, the explainable function can help determine input parameters that affect each output value in the bottom graph.

Similar to Figures 6.20 and 6.21, organic loading, BOD_5, and flow rate were the most related parameters to SVI prediction, followed by TKN, TP, NH_3N, and TSS. Depending on the operation condition, the principal parameters affecting the prediction varied. Therefore, applying this explainable function along with model prediction would assist the WWTP operation by closely monitoring the system, visualizing and controlling the system, making a model prediction, interpreting the result, and providing a faulty alarm.

```
Test RMSE: 3.357
Test MAE: 2.347
```

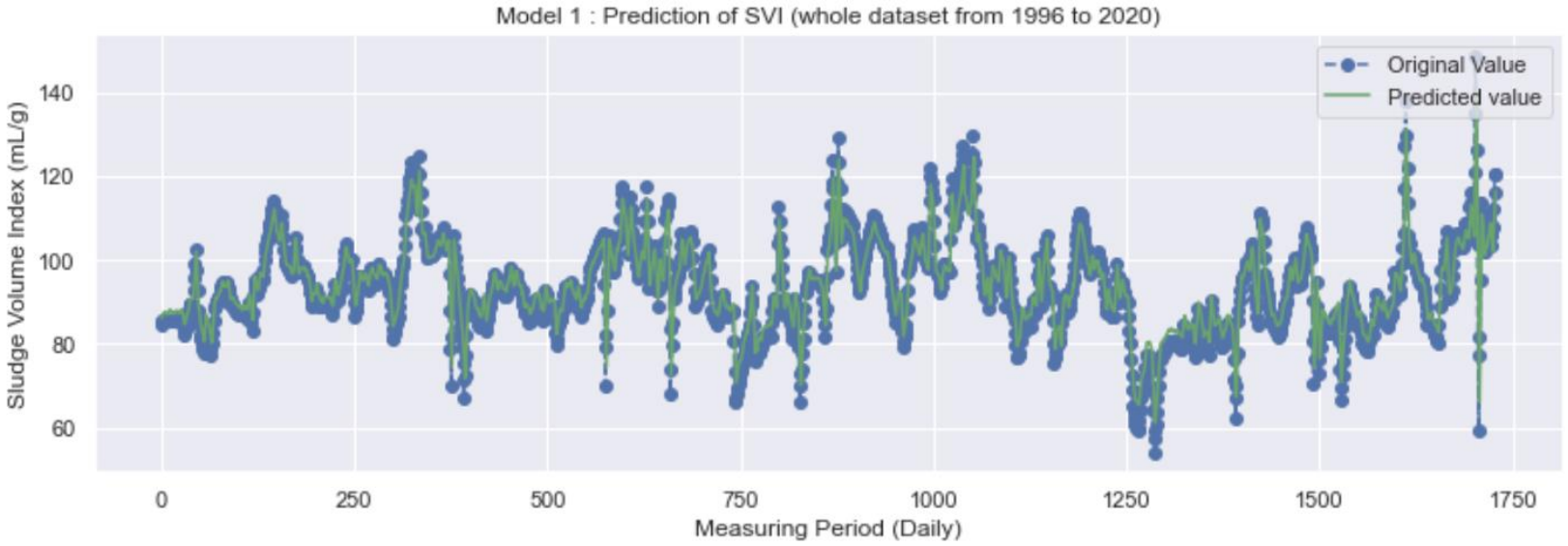


Figure 6.16 The Sludge Volume Index prediction model of the first set of data (1996 to 2020).

```
Test RMSE: 3.076
Test MAE: 2.163
```

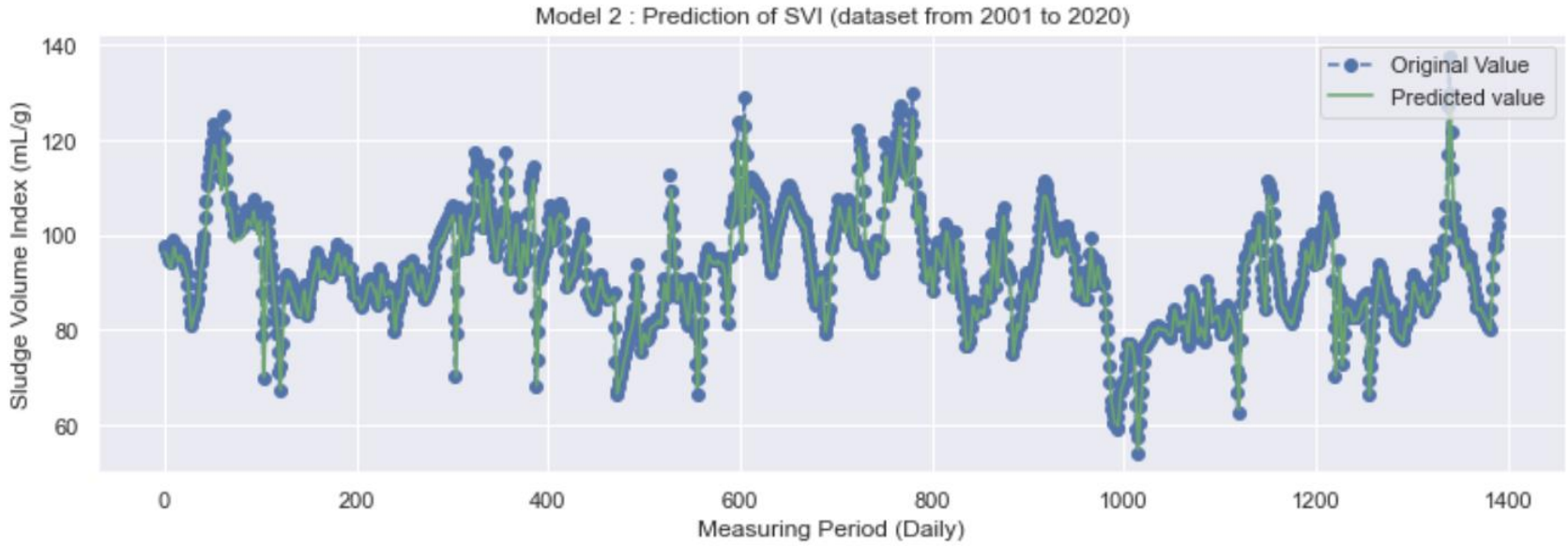


Figure 6.17 The Sludge Volume Index prediction model of the second set of data (2001 to 2020).

```
Test RMSE: 3.061
Test MAE: 2.406
```

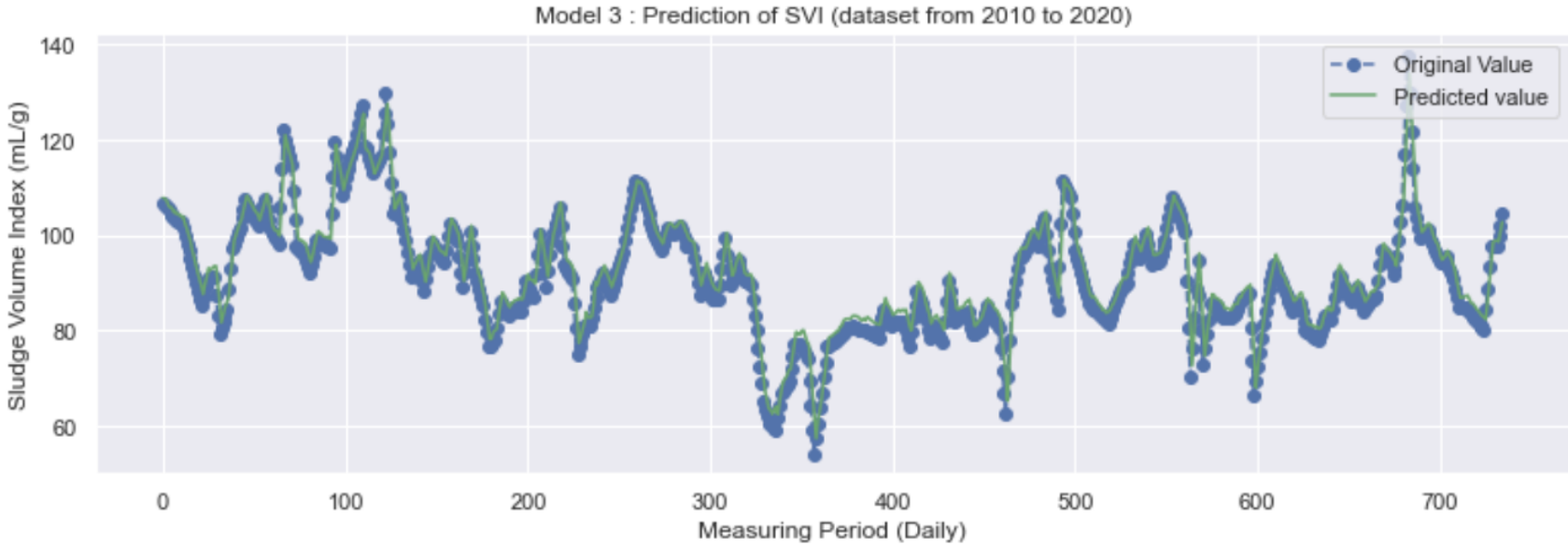


Figure 6.18 The Sludge Volume Index prediction model of the third set of data (2010 to 2020).

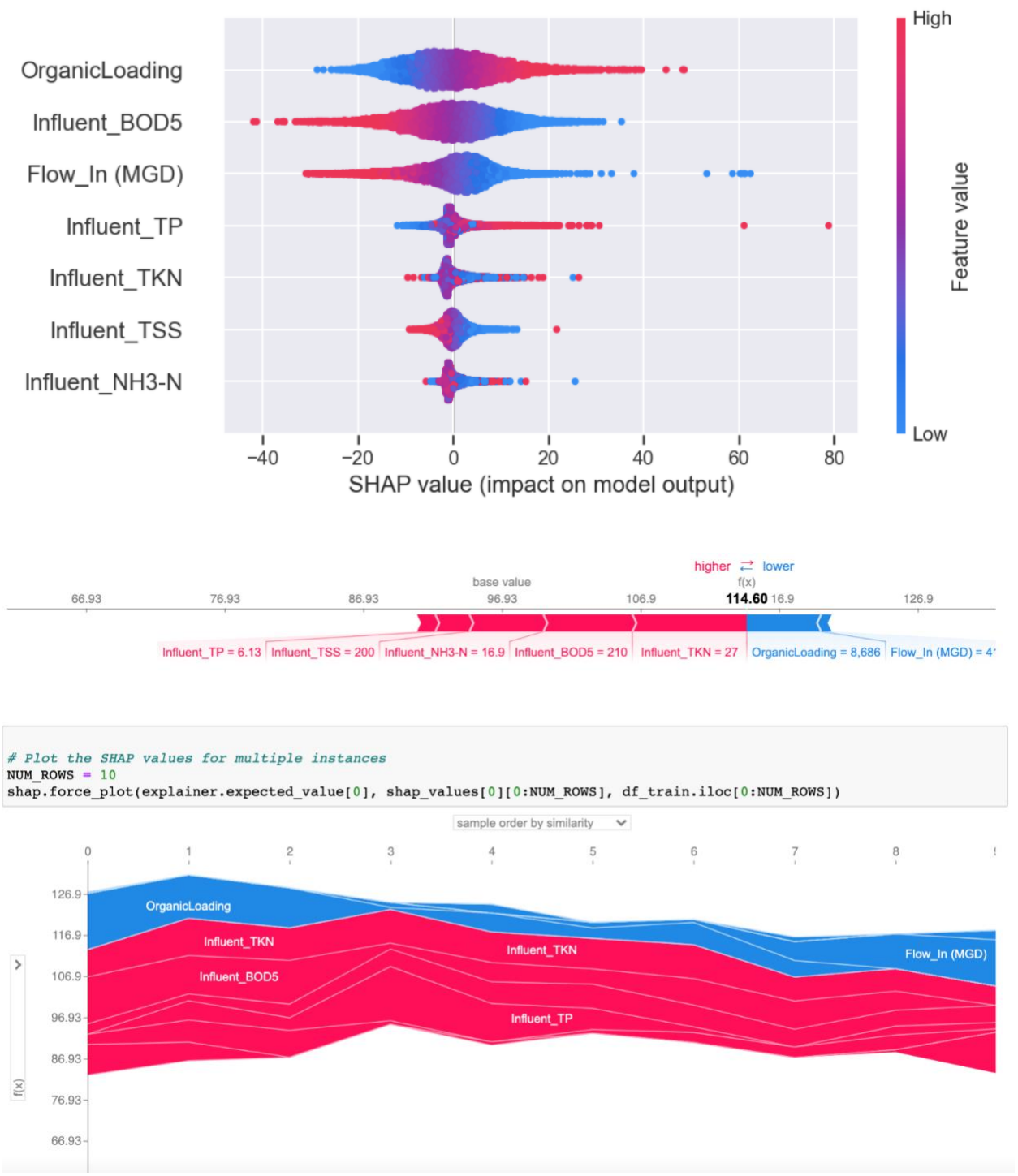


Figure 6.19 SHapley interpretation plots for the first model.

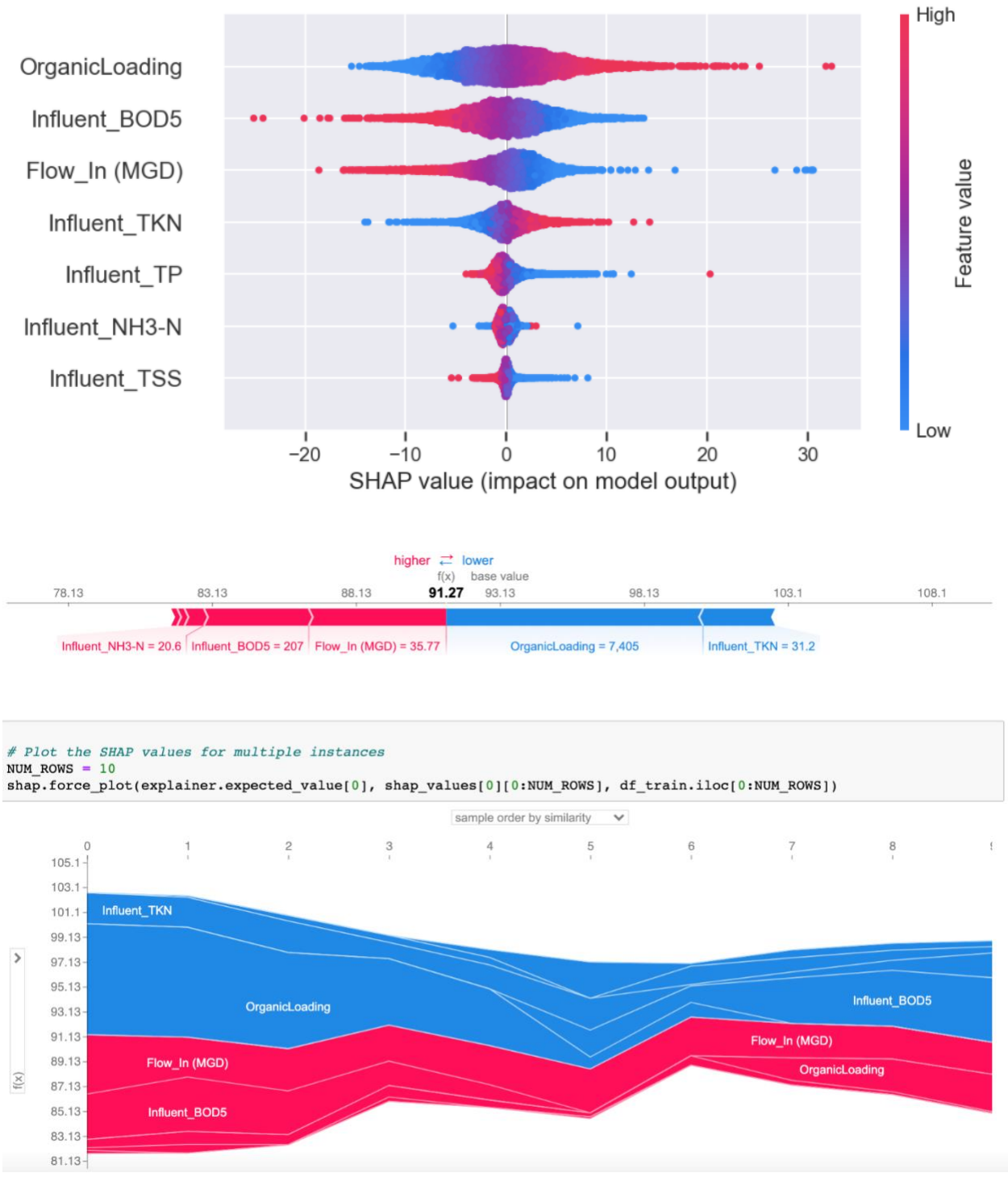


Figure 6.20 SHapley interpretation plots for the second model.

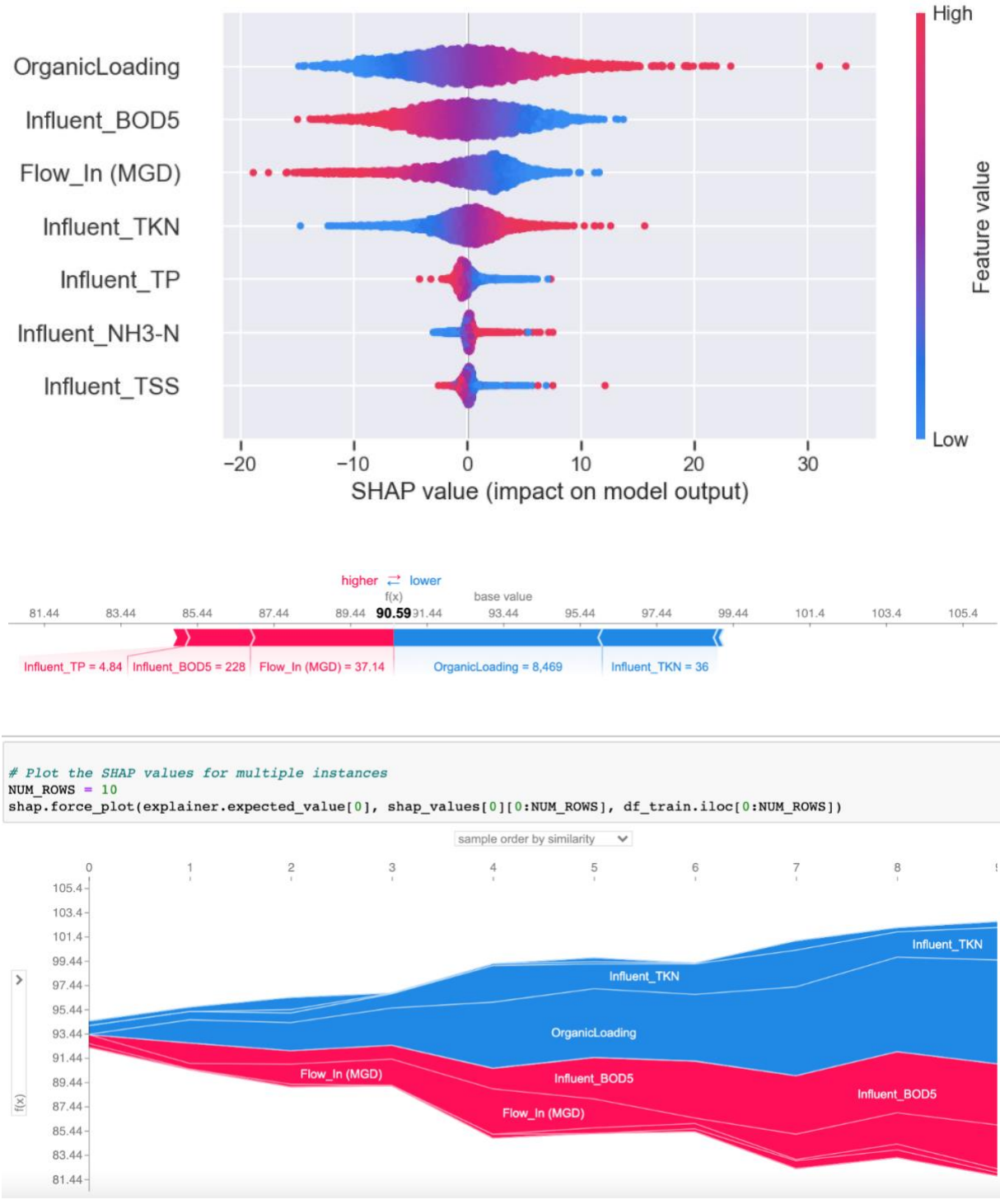


Figure 6.21 SHapley interpretation plots for the third model.

6.5 Conclusions and Recommendations

The activated sludge process is one of the most widely used wastewater treatment processes. SVI is the important operational parameter to determine the solid separation potential in the activated sludge system. The RNN model for SVI prediction with data analysis and an Explainable AI function was found to be suitable for WWTP operation. The first step is to collect the data from the WWTP, followed by the data analysis and visualization to see the data pattern. The second step is to select an appropriate dataset for an RNN model. The last step is to apply the interpretation method for explaining the prediction result and find a root cause of the problems.

The results showed the model's ability to predict SVI despite significant fluctuation in the dataset. The RMSE was ~3 and MAE was ~2 in the SVI range from 50 to 150. However, the RMSE and MAE are still not low enough to conclude that a model is good because it is scale-dependent. A good model can be determined by looking at training and testing errors. The result shows that the second and third models were an excellent in-sample fit, associated with low error measures.

The Explainable Artificial Intelligence (AI) algorithm was useful in explaining the causes of high SVI values. The result can be interpreted in the graph and shows what causes SVI higher or lower for each instance and what the most related parameters to output prediction are. Therefore, it can be concluded that an RNN model with explainable AI can successfully predict SVI and suggest operators which parameters affect higher SVI under widely varying daily conditions.

From the Nine Springs WWTP data, SVI was found to be affected most by organic loading and thus influent BOD_5 and flow rate. Moreover, as discussed in the previous chapter, aeration control should be thoroughly monitored in an aeration system because it significantly impacts the effluent BOD_5 and SVI. Although it was possible to determine which parameter(s) caused higher SVI,

reasons and corrective measures must be investigated further for individual WWTPs. In addition, the logical investigation method must be developed and validated.

7. REFERENCES

Aerzen. (2021). *White Paper: Aeration Blowers In The Wastewater Industry In North America*. Aerzen USA Corporation. www.aerzen.com

Bakia, O. T., Arasb, E., Akdemirc, U. O., & Yilmaza, B. (2019). *Biochemical oxygen demand prediction in WWTP by using different regression analysis models.*

Boyd, G., Na, D., Li, Z., Snowling, S., Zhang, Q., & Zhou, P. (2019). Influent Forecasting for Wastewater Treatment Plants in North America. *Sustainability*.

Brownlee, J. (2017). *Long Short-Term Memory Networks With Python* (v 1.0). https://machinelearningmastery.com/lstms-with-python/

Brownlee, J. (2019, February 27). *How to use Learning Curves to Diagnose Machine Learning Model Performance*. Machine Learning Mastery. https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

Bye, C., & Dold, P. (1998). Sludge volume index settleability measures: Effect of solids characteristics and test parameters. *Water Environment Research*, *70*, 87–93. https://doi.org/10.2175/106143098X126928

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

Cheng, T., Dairi, A., Harrou, F., Sun, Y., & Leiknes, T. (2019). Monitoring Influent Conditions of Wastewater Treatment Plants by Nonlinear Data-Based Techniques. *IEEE Access*, *7*, 108827–108837. https://doi.org/10.1109/ACCESS.2019.2933616

Choi, E., Bahadori, M. T., Kulas, J. A., Schuetz, A., Stewart, W. F., & Sun, J. (2017). RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. *ArXiv:1608.05745 [Cs]*. http://arxiv.org/abs/1608.05745

D'agostino, R. B., Belanger, A., & Jr, R. B. D. (1990). A Suggestion for Using Powerful and Informative Tests of Normality. *The American Statistician*, *44*(4), 316–321. https://doi.org/10.1080/00031305.1990.10475751

Daw, J., Hallett, K., DeWolfe, J., & Venner, I. (2012). *Energy Efficiency Strategies for Municipal Wastewater Treatment Facilities* (NREL/TP-7A20-53341, 1036045; p. NREL/TP-7A20-53341, 1036045). https://doi.org/10.2172/1036045

Dochain, D., & Vanrolleghem, P. A. (2000). Dynamical Modelling & Estimation in Wastewater Treatment Processes | Request PDF. *ResearchGate*. https://www.researchgate.net/publication/243769834_Dynamical_Modelling_Estimation_in_Wastewater_Treatment_Processes

Du, J., Kuang, B., & Yang, Y. (2019). A Data-Driven Framework for Smart Urban Domestic Wastewater: A Sustainability Perspective. *Advances in Civil Engineering*, *2019*, e6530626. https://doi.org/10.1155/2019/6530626

Durrenmatt, D. J. (2011). *Data mining and data-driven modeling approaches to support WWTP operation*.

Fernando, J. (2020, November 18). *R-Squared*. Investopedia. https://www.investopedia.com/terms/r/r-squared.asp

Few, S. (2014). *Data Visualization for Human Perception* (2nd ed.). https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception

Finnegan, T. (2020). Operation and Control. *Environmental Leverage Inc.*, 12.

Frankenfield, J. (2020, May 12). *Artificial Neural Network (ANN)*. Investopedia. https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp

Garbowski, T., Wiśniewski, J., & Bawiec, A. (2018). Analysis and Assesment of the Wastewater Treatment Plant Operation in the City of Kłodzko. *Journal of Ecological Engineering*, *19*(2), 114–124. https://doi.org/10.12911/22998993/81785

Ghernaout, D., Aichouni, M., & Alghamdi, A. (2018). Applying big data in water treatment industry: A new era of advance. *International Journal of ADVANCED AND APPLIED SCIENCES*, *5*(3), 89–97. https://doi.org/10.21833/ijaas.2018.03.013

Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of Neural Networks Is Fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 3681–3688. https://doi.org/10.1609/aaai.v33i01.33013681

Grant, S. B., Saphores, J.-D., Feldman, D. L., Hamilton, A. J., Fletcher, T. D., Cook, P. L. M., Stewardson, M., Sanders, B. F., Levin, L. A., Ambrose, R. F., Deletic, A., Brown, R., Jiang, S. C., Rosso, D., Cooper, W. J., & Marusic, I. (2012). Taking the "waste" out of "wastewater" for human water security and ecosystem sustainability. *Science (New York, N.Y.)*, *337*(6095), 681–686. https://doi.org/10.1126/science.1216852

Hagquist, C., & Stenbeck, M. (1998). Goodness of Fit in Regression Analysis – R2 and G2 Reconsidered. *Quality and Quantity*, *32*(3), 229–245. https://doi.org/10.1023/A:1004328601205

Haider, S. A., Naqvi, S. R., Akram, T., Umar, G. A., Shahzad, A., Sial, M. R., Khaliq, S., & Kamran, M. (2019). LSTM Neural Network Based Forecasting Model for Wheat Production in Pakistan. *Agronomy*, *9*(2), 72. https://doi.org/10.3390/agronomy9020072

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). https://www.elsevier.com/books/data-mining-concepts-and-techniques/han/978-0-12-381479-1

Harrou, F., Dairi, A., Sun, Y., & Senouci, M. (2018). Statistical monitoring of a WWTP: A case study. *Journal of Environmental Management*, *223*, 807–814. https://doi.org/10.1016/j.jenvman.2018.06.087

Henze, M., Gujer, W., Mino, T., & van Loosedrecht, M. (2015). Activated Sludge Models ASM1, ASM2, ASM2d and ASM3. *Water Intelligence Online*, *5*(0), 9781780402369–9781780402369. https://doi.org/10.2166/9781780402369

Hiregoudar, S. (2020, August 4). *Ways to Evaluate Regression Models*. Medium. https://towardsdatascience.com/ways-to-evaluate-regression-models-77a3ff45ba70

Huang, M., Han, W., Wan, J., Ma, Y., & Chen, X. (2016). Multi-objective optimisation for design and operation of anaerobic digestion using GA-ANN and NSGA-II. *Journal of Chemical Technology & Biotechnology*, *91*(1), 226–233. https://doi.org/10.1002/jctb.4568

Jenkins, D., Richard, M. G., & Daigger, G. T. (2003). *Manual on the Causes and Control of Activated Sludge Bulking, Foaming, and Other Solids Separation Problems* (3rd edition). CRC Press.

Khademikia, S., Haghizadeh, A., Godini, H., & Khorramabadi, G. S. (2016). Artificial Neural Network-Cuckoo Optimization Algorithm (ANN-COA) for Optimal Control of Khorramabad Wastewater Treatment Plant, Iran. *Civil Engineering Journal*, *2*(11), 555-567–567. https://doi.org/10.28991/cej-2016-00000058

Krejcik, J. (2020, April 2). The one tip for an efficient WWTP: Go digital. *Blog*. https://blog.dhigroup.com/2020/04/02/the-one-tip-for-an-efficient-wastewater-treatment-plant-go-digital/

Lamons, M., Kumar, R., & Nagaraja, A. (2018). *Python Deep Learning Projects: 9 projects demystifying neural network and deep learning models for building intelligent systems*. Packt Publishing Ltd.

Li, S. (2019, May 16). *Time Series Analysis, Visualization & Forecasting with LSTM*. https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-77a905180eba

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *ArXiv:1705.07874 [Cs, Stat]*. http://arxiv.org/abs/1705.07874

Maiza, M., Beltrán, S., Westling, K., Carlsson, B., Mulas, M., Bergström, P.-H., Hyyryläinen, S.-M., & Urchegui, G. (2013, September). *DIAMOND: AdvanceD data management and InformAtics for the optimuM operatiON anD control of WWTPs*. 11th IWA Conference on Instrumentation Control and Automation, Narbonne, France. https://www.researchgate.net/publication/253650390_DIAMOND_AdvanceD_data_management_and_InformAtics_for_the_optimuM_operatiON_anD_control_of_WWTPs#fullTextFileContent

Man, Y., Hu, Y., & Ren, J. (2019). Forecasting COD load in municipal sewage based on ARMA and VAR algorithms. *Resources, Conservation, and Recycling*, *144*(1), 56–64.

Manda, V. K., Poosapati, V., & Katneni, V. (2018). *Super SCADA Systems: A Prototype for Next Gen SCADA System*.

Martin, C., & Vanrolleghem, P. A. (2014). Analysing, completing, and generating influent data for WWTP modelling: A critical review. *Environmental Modelling & Software*, *60*, 188–201. https://doi.org/10.1016/j.envsoft.2014.05.008

Mauro, A. D., Greco, M., & Grimaldi, M. (2015). *What is big data? A consensual definition and a review of key research topics*. 97–104. https://doi.org/10.1063/1.4907823

McGowan, S., & Wang, E. (2008). *50-Year Master Plan Review of Existing Treatment Facilities*. Madison MSD Project No. 8425001 Malcolm Pirnie Project No. 6100-001.

Metcalf & Eddy, Tchobanoglous, G., Franklin L. Burton, & Stensel, H. D. (Eds.). (2003). *Wastewater engineering: Treatment and reuse* (4th ed). McGraw-Hill.

Millinger, A. (2020, July 22). *The Modernization of SCADA and HMI*. Water & Wastes Digest. https://www.wwdmag.com/scada-systems/modernization-scada-and-hmi

Morsi, I., Deeb, M. E., & Zwawi, A. E. (2009). SCADA/HMI Development for a Multi Stage Desalination Plant. *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, 67–71. https://doi.org/10.1109/ComputationWorld.2009.114

Nadiri, A. A., Shokri, S., Tsai, F. T.-C., & Moghaddam, A. A. (2018). Prediction of Effluent Quality Parameters of a Wastewater Treatment Plant Using a Supervised Committee Fuzzy Logic Model. *Journal of Cleaner Production 180*, *180*, 539–549.

Najafzadeh, M., & Zeinolabedini, M. (2019). Prognostication of waste water treatment plant performance using efficient soft computing models: An environmental evaluation. *Measurement*, *138*, 690–701. https://doi.org/10.1016/j.measurement.2019.02.014

Newhart, K. B., Holloway, R. W., Hering, A. S., & Cath, T. Y. (2019). Data-driven performance analyses of WWTPs: A review. *Water Research*, *157*, 498–513. https://doi.org/10.1016/j.watres.2019.03.030

Nicholson, C. (2020). *Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning*. Pathmind. http://pathmind.com/wiki/ai-vs-machine-learning-vs-deep-learning

Ning, R. (2018). Sewage Treatment Process Based on Big Data Management Mode. *Chemical Engineering Transactions*, *71*, 703–708. https://doi.org/10.3303/CET1871118

Olah, C. (2015, August 27). *Understanding LSTM Networks—Colah's blog*. Colah's Blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pantsar-Kallio, M., Mujunen, S.-P., Hatzimihalis, G., Koutoufides, P., Minkkinen, P., Wilkie, P. J., & Connor, M. A. (1999). Multivariate data analysis of key pollutants in sewage samples: A case study. *Analytica Chimica Acta*, *393*(1–3), 181–191.

Phuc, B. D. H., You, S.-S., Hung, B. M., & Kim, H.-S. (2018). Robust control synthesis for the activated sludge process. *Environmental Science: Water Research & Technology*, *4*(7), 992–1001. https://doi.org/10.1039/C8EW00032H

Pisa, I., Santín, I., Vicario, J. L., Morell, A., & Vilanova, R. (2019). ANN-Based Soft Sensor to Predict Effluent Violations in Wastewater Treatment Plants. *Sensors*, *19*(6), 1280. https://doi.org/10.3390/s19061280

Qiao, J. F., Yang, W. W., & Yuan, M. Z. (2011). Recurrent High Order Neural Network Modeling for Wastewater Treatment Process. *Journal of Computers*, *6*(8), 1570–1577. https://doi.org/10.4304/jcp.6.8.1570-1577

Riahi, Y., & Riahi, S. (2018). Big Data and Big Data Analytics: Concepts, types and technologies. *International Journal of Research and Engineering*, *5*(9), 524–528.

Richards, R. (2020, April 6). *Novel methods for monitoring wastewater from the wastewater network and across the sewage treatment plant to aid optimisation*. Environmental Technology. https://www.envirotech-online.com/article/water-wastewater/9/swig/novel-methods-for-monitoring-wastewater-from-the-wastewater-network-and-across-the-sewage-treatment-plant-to-aid-optimisation/2727

Ritchie, H., & Roser, M. (2018). Water Use and Stress. *Our World in Data*. https://ourworldindata.org/water-use-stress

Romero, J. M. P., Hallett, S. H., & Jude, S. (2017). Leveraging Big Data Tools and Technologies: Addressing the Challenges of the Water Quality Sector. *Sustainability*, *9*(12), 2160. https://doi.org/10.3390/su9122160

Roser, M., Ritchie, H., & Ortiz-Ospina, E. (2015). Internet. *Our World in Data*. https://ourworldindata.org/internet

Rosso, D., Stenstrom, M. K., & Larson, L. E. (2008). Aeration of large-scale municipal WWTPs: State of the art. *Water Science and Technology: A Journal of the International Association on Water Pollution Research*, *57*(7), 973–978. https://doi.org/10.2166/wst.2008.218

Shaw, A. (2017, March 20). *Understanding Big Data In The Water Industry*. https://www.wateronline.com/doc/understanding-big-data-in-the-water-industry-0002

Shi, S., & Xu, G. (2018). Novel performance prediction model of a biofilm system treating domestic wastewater based on stacked denoising auto-encoders deep learning network. *Chemical Engineering Journal*, *347*, 280–290. https://doi.org/10.1016/j.cej.2018.04.087

Siegrist, R. L. (2017). Introduction to Decentralized Infrastructure for Wastewater Treatment and Water Reclamation. In R. L. Siegrist (Ed.), *Decentralized Water Reclamation Engineering:*

A Curriculum Workbook (pp. 1–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-40472-1_1

Sosik, S. J. (2014). *SCADA SYSTEMS IN WASTEWATER TREATMENT*. Jomo Kenyatta University of Agriculture and Technology.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *ArXiv:1409.3215 [Cs]*. http://arxiv.org/abs/1409.3215

Taheriyoun, M., & Moradinejad, S. (2015). Reliability analysis of a WWTP using fault tree analysis and Monte Carlo simulation. *Environmental Monitoring and Assessment*, *187*(1), 4186. https://doi.org/10.1007/s10661-014-4186-7

Tesh, K. (2016, July 7). HOW TO CALCULATE SLUDGE VOLUME INDEX - SVI. *Water and Wastewater Courses*. https://www.waterandwastewatercourses.com/calculate-sludge-volume-index-svi/

U.S. EPA. (2013). *Energy Efficiency in Water and Wastewater Facilities: A Guide to Developing and Implementing Greenhouse Gas Reduction Programs*. 56.

Wei, X. (2013). *Modeling and optimization of wastewater treatment process with a data-driven approach* [Doctor of Philosophy, University of Iowa]. https://doi.org/10.17077/etd.wwzj01nf

Wikipedia. (2020). Sludge volume index. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Sludge_volume_index&oldid=963975303

Wu, J., Cheng, Y., & Schulz, N. N. (2006). Overview of Real-Time Database Management System Design for Power System SCADA System. *Proceedings of the IEEE SoutheastCon 2006*, 62–66. https://doi.org/10.1109/second.2006.1629324

Yousuf, I. (2013). *METHODS FOR ESTIMATION AND COMPARISON OF ACTIVATED SLUDGE SETTLEABILITY*. 7.

Yu, P., Cao, J., Jegatheesan, V., & Shu, L. (2019). Activated sludge process faults diagnosis based on an improved particle filter algorithm. *Process Safety and Environmental Protection*, *127*, 66–72. https://doi.org/10.1016/j.psep.2019.04.021

Zhang, Q., Li, Z., Snowling, S., Siam, A., & El-Dakhakhni, W. (2019). Predictive models for wastewater flow forecasting based on time series analysis and artificial neural network. *Water Science and Technology*, *80*(2), 243–253. https://doi.org/10.2166/wst.2019.263

Zhao, L., Dai, T., Qiao, Z., Sun, P., Hao, J., & Yang, Y. (2020). Application of artificial intelligence to wastewater treatment: A bibliometric analysis and systematic review of technology, economy, management, and wastewater reuse. *Process Safety and Environmental Protection*, *133*, 169–182. https://doi.org/10.1016/j.psep.2019.11.014

8. APPENDICES

Table 8.1 Application of AI technologies to pollutant removal in WWTPs.

Item	No.	Simulation or Prediction Objective	Treatment Process	AI Model	Training data sets/%	Validation data sets/%	Testing data sets/%	Model Performance	Reference
Conventional pollutant	1	COD	Aeration, nitrification & denitrification	ANN	75	–	25	0.632[a]	(Moral et al., 2008)
	2	COD	Anoxic oxic biological	ANFIS	70	18	12	0.982[a]	(Wan et al., 2011)
	3	COD	Anaerobic digestion	ANN-GA ANN	70	30	–	196.1[b] 447.7[b]	(Huang et al., 2016)
	4	COD	Fenton oxidation	MLP-ANN	61	17	22	0.975[a]	(Sabour and Amiri, 2017)
	5	COD	Aeration diffusion	SCFL	80	–	20	0.950[a]	(Nadiri et al., 2018)
	6	COD	Aeration diffusion	ARMA-VAR BP-ANN GA-BP-ANN	95	–	5	113.56 [b] 303.51[b] 232.6 [b]	(Man et al., 2019)
	7	COD	Activated sludge	GM-GA				0.85[a]	(Chen et al., 2010)
	8	BOD_5	Aeration diffusion	ANN	70	–	30	0.810[a]	(Hamed et al., 2004)
	9	BOD_5	Aeration diffusion	SCFL FL	81	–	20	0.960[a] 0.820[a]	(Nadiri et al., 2018)
	10	BOD_5	Biological	NNE	–	–	–	20 %[c]	(Nourani et al., 2018)
	11	BOD_5	Activated sludge bioreactor	ARIMA-ORELM	–	–	–	0.990[a]	(Lotfi et al., 2019)
	12	NH_4^+, NO_3-	Contact aeration	BP-ANN	–	–	–	90 %[d]	(Chen et al., 2003a,2003b)
	13	PO_4^{3-}	Sequencing batch reactor	ABM	–	–	–	0.580[b]	(Bucci et al., 2012)
	14	NO_3-	Biochemical	MOPSO	–	–	–	0.034[e]	(Han et al., 2018a, 2018b)
	15	TN	Biological	NNE	–	–	–	5 %[c]	(Nourani et al., 2018)
	16	PO_4^{3-}	Anaerobic and aerobic	Q-learning	–	–	–	–	(Pang et al., 2019)
	17	Total inorganic nitrogen	Arobic activated sludge	HMMs-MNLR	–	–	–	84 %[d]	(Suchetana et al., 2019)
	18	NH_4^+, TN	Anammox & Partial nitritation	FFBP-ANN	70	15	15	0.997[a]	(Antwi et al., 2019a, 2019b)
	19	TN TP	Sequencing batch reactor	BN	–	–	–	93.1 %[d] 95.2 %[d]	(Li et al., 2013)
	20	PO_4^{3-}	Adsorption	BP-ANN-GA	68	22	11	0.990[a]	(Zhang and Pan, 2014)
	21	Cu^{2+}	Membrane	RBF-ANN	–	–	–	0.997[a]	(Messikh et al., 2015)
	22	Cd^{2+}	Adsorption	ANFIS	–	–	–	0.921[a]	(Fawzy et al., 2016)
	23	As^{3+} As^{5+}	Phytoremediation	ANN	60	20	20	1.000[a] 1.000[a]	(Podder and Majumder, 2016)
	24	As^{3+}	Adsorption	BP-ANN	59	–	42	0.980[a]	(Mandal et al., 2015)
	25	Pb^{2+}	Adsorption	BP-ANN	70	15	15	0.970[a]	(Reynel-Avila et al., 2015)
	26	Pb^{2+}	Adsorption	MLP-ANN RSM	–	–	–	0.990[a] 0.956[a]	(Peiman et al., 2019)
	27	Mn^{2+}	Extraction	BP-ANN-PSO	76	12	12	0.981[a]	(Khajeh and Barkhordar, 2013)
	28	Cr^{6+}	Adsorption	BP-ANN-GA	77	–	23	0.995[a]	(Mohan et al., 2015)
Typical pollutant	29	Naphthalene	Photodegradation	ANN	60	20	20	0.943[a]	(Jing et al., 2014)
	30	Methylene blue	Photocatalytic	RBF-ANN	–	–	–	–	(Ranjbar-mohammadi et al., 2019)

L. Zhao, T. Dai, Z. Qiao et al. / Process Safety and Environmental Protection 133 (2020) 168–182

(Zhao et al., 2020)

Table 8.2 Application of AI technologies to pollutant removal in WWTPs (Continued).

Application of AI models for operation management during wastewater treatment.

No.	Operation Management objective	Treatment Process	AI Model	Training data sets/%	Validation data sets/%	Testing data sets/%	Performance	Reference
1	Aeration efficiency	Aerobic biological	ANFIS ANN	78	–	22	0.99[a] 0.95[a]	(Huang et al., 2009)
2	Aeration efficiency	Aeration or reaeration	EPR MT	75	–	25	0.88[a] 0.93[a]	(Sattar et al., 2019)
3	Anaerobic system	Anaerobic fluidized bed Anaerobic filter Upflow anaerobic sludge	NF	–	–	–	0.82[a] 0.92[a] 0.84[a]	(Tay and Zhang, 2000)
4	Pump system	Typical treatment process	DM	80	–	20	0.93[a]	(Zhang et al., 2016)
5	Sludge bulking	Sequencing batch reactor	MLP-ANN RBF-ANN	70	–	30	0.99[a] 0.96[a]	(Bagheri et al., 2015)
6	Sludge bulking	Activated sludge	Information-oriented algorithm	57	–	43	1.0 %[b]	(Han et al., 2018a)
7	Sludge bulking	Activated sludge	Mard-RCP	60	–	40	–	(Cheng et al., 2019)
8	Activated sludge process fault	Activated sludge	PFA	–	–	–	0.25[c]	(Yu et al., 2019)
9	Permeate flux	Microfiltration	FFNN	67	–	33	0.99[a]	(Aydiner et al., 2005)
10	Membrane fouling	Cross-flow microfiltration	MLP-ANN	63	–	37	–	(Dornier et al., 1995)
11	Permeate flux	Membrane bioreactor	RBF-ANN RBF-ANN-GA MLP-ANN MLP-ANN-GA	67		33	0.99[a] 0.99[a] 0.99[a] 0.99[a]	(Schmitt and Do, 2017)
12	Membrane fouling	Membrane bioreactor	ANN/SVM/RNN/ENN/WNN/SOM	–	–	–	–	(Bagheri et al., 2019)
13	Membrane fouling	Membrane bioreactor	RBF-ANN	50	28	22	2.23 %[d]	(Chen et al., 2019)
14	Membrane fouling	Membrane bioreactor	Least squares SVM	80	–	20	0.99 [a]	(Hamedi et al., 2019)
15	Daily flow rate	Activated sludge	SVM BP-ANN ANFIS RBF-ANN	70	–	30	1435.4[c] 1445.9[c] 1515.6[c] 1501[c]	(Najafzadeh and Zeinolabedini, 2019)
16	Influent flow rate	Wastewater treatment	ANFIS-GWO ANFIS	70	–	30	0.98[a] 0.97[a]	(Dehghani et al., 2019)
17	Monitoring control	Oxidation–reduction & neutralization	SOM	–	–	–	–	(Garĉa and González, 2004)
18	Sensor control	Anaerobic Aerobic	NF BP-ANN NF BP-ANN	80	10	10	0.96[a] 0.90[a] 0.98[a] 0.95[a]	(Huang et al., 2010)
19	Automated control	Activated sludge	RL	–	–	–	–	(Hernández-Del-Olmo et al., 2012)
20	Automated control	Biological reactor	Hybrid AI	–	–	–	–	(Wen and Vassiliadis, 2002)
21	Systematic control	Activated sludge	MOOC	–	–	–	–	(Dai et al., 2016)

a: Determination coefficient (R^2); b: Accuracy; c: Root mean square error (RMSE); d: Relative error.

Table 8.3 Application of AI models for operation management during wastewater treatment.

Item	No.	Simulation or Prediction Objective	Treatment Process	AI Model	Training data sets/%	Validation data sets/%	Testing data sets/%	Model Performance	Reference
Mixed pollutant	31	Methylene blue	Adsorption	MLP-ANN	70	15	15	0.988[a]	(Asfaram et al., 2017)
				RBF-ANN				0.999[a]	
	32	Tris-styrene	Adsorption	BP-ANN-GA	45	–	19	0.986[a]	(Ghaedi et al., 2016)
				MLR	–	–	–	0.751[a]	
	33	Trichlorophenol	Adsorption	MLP-ANN	–	–	–	0.999[a]	(Dlamini et al., 2014)
	34	Methylene blue	Adsorption	BP-ANN	70	15	15	1.000[a]	(Asfaram et al., 2016)
		Malachite green						0.999[a]	
	35	Malachite green	Adsorption	BP-ANN-GA	75	–	25	0.966[a]	(Ghaedi et al., 2015)
	36	Bisphenol A	Adsorption	ANN	60	20	60	0.997[a]	(Vakili et al., 2019)
				RSM				0.995[a]	
		Carbamazepine		ANN				0.997[a]	
				RSM				0.984[a]	
		Ketoprofen		ANN				0.998[a]	
				RSM				0.998[a]	
		Tonalide		ANN				0.993[a]	
				RSM				0.994[a]	
	37	Boron	Electrocoagulation	ANN	80	–	20	0.973[a]	(da Silva Ribeiro et al., 2019)
	38	Microbead	Physico-chemical	CNN	75	–	25	89 %[d]	(Yurtsever and Yurtsever, 2019)
	39	Zn^{2+}, Cu^{2+}, Pb^{2+}, Cd^{2+}, Ni^{2+}, As^{3+}	Adsorption	ANN	9	–	1	0.948[a]	(Zhu et al., 2019)
				Random forest				0.973[a]	
	40	COD	Sequencing batch reactor	MLP-ANN	70 70	15	15	0.990[a]	(Bagheri et al., 2015)
				RBF-ANN		–	30	0.990[a]	
		NH_4^+		MLP-ANN	70	15	15	0.990[a]	
				RBF-ANN	70	–	30	0.990[a]	
		TP		MLP-ANN	70	15	15	0.990[a]	
				RBF-ANN	70	–	30	0.940[a]	
	41	COD	Anaerobic oxic biological	SDAE	81	–	19	5.94[b]	(Shi and Xu, 2018)
		NH_4^+						1.27[b]	
		TN						1.26[b]	
	42	COD	Sequencing batch reactor	BP-ANN	52	24	24	0.955[a]	(Kundu et al., 2013)
		NH_4^+						0.958[a]	
	43	Methylene blue	Adsorption	BP-ANN	69	15	15	0.990[a]	(Mazaheri et al., 2017)
				BRT				0.992[a]	
		Cd^{2+}		BP-ANN				0.989[a]	
				BRT				0.991[a]	
	44	Pb^{2+}	Adsorption	MLP-ANN	63	19	19	1.000[a]	(Dil et al., 2017)
		Malachite green						1.000[a]	

a: Determination coefficient (R^2); b: Root mean square error (RMSE); c: Performance efficiency; d: Accuracy; e: Integral of the squared error.

L. Zhao, T. Dai, Z. Qiao et al. / Process Safety and Environmental Protection 133 (2020) 169–182

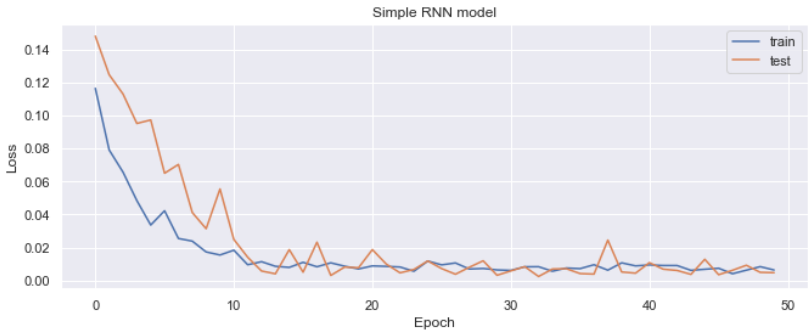
(Zhao et al., 2020)

```
In [8]: model = Sequential()
        model.add(SimpleRNN(50, input_shape=(train_X.shape[1], train_X.shape[2])))
        model.add(Dense(1))

In [9]: model.compile(loss='mae', optimizer='adam')
        history = model.fit(train_X, train_y, epochs=50, batch_size=100,
                            validation_data=(test_X, test_y), verbose=2, shuffle=False)
Train on 8760 samples, validate on 82766 samples
Epoch 1/50
 - 2s - loss: 0.1164 - val_loss: 0.1480
Epoch 2/50
 - 1s - loss: 0.0791 - val_loss: 0.1246
Epoch 3/50
 - 1s - loss: 0.0656 - val_loss: 0.1130
Epoch 4/50
 - 1s - loss: 0.0484 - val_loss: 0.0951
Epoch 5/50
 - 1s - loss: 0.0336 - val_loss: 0.0972
Epoch 6/50
 - 1s - loss: 0.0422 - val_loss: 0.0650
Epoch 7/50
 - 1s - loss: 0.0254 - val_loss: 0.0703
Epoch 8/50
 - 1s - loss: 0.0238 - val_loss: 0.0412
Epoch 9/50
 - 1s - loss: 0.0173 - val_loss: 0.0313
Epoch 10/50
 - 1s - loss: 0.0154 - val_loss: 0.0554
Epoch 11/50
 - 1s - loss: 0.0184 - val_loss: 0.0249
Epoch 12/50
 - 1s - loss: 0.0095 - val_loss: 0.0141
Epoch 13/50
 - 1s - loss: 0.0114 - val_loss: 0.0057
Epoch 14/50
 - 1s - loss: 0.0086 - val_loss: 0.0040
Epoch 15/50
 - 1s - loss: 0.0079 - val_loss: 0.0187
Epoch 16/50
 - 1s - loss: 0.0110 - val_loss: 0.0050
Epoch 17/50
 - 1s - loss: 0.0083 - val_loss: 0.0232
Epoch 18/50
 - 1s - loss: 0.0107 - val_loss: 0.0031
Epoch 19/50
 - 1s - loss: 0.0087 - val_loss: 0.0081
Epoch 20/50
 - 1s - loss: 0.0070 - val_loss: 0.0077
Epoch 21/50
 - 1s - loss: 0.0088 - val_loss: 0.0187
Epoch 22/50
 - 2s - loss: 0.0086 - val_loss: 0.0100
Epoch 23/50
 - 1s - loss: 0.0082 - val_loss: 0.0046
```

Figure 8.1 A simple RNN model in Python.

```
Epoch 24/50
 - 1s - loss: 0.0056 - val_loss: 0.0067
Epoch 25/50
 - 1s - loss: 0.0118 - val_loss: 0.0118
Epoch 26/50
 - 2s - loss: 0.0095 - val_loss: 0.0072
Epoch 27/50
 - 2s - loss: 0.0106 - val_loss: 0.0038
Epoch 28/50
 - 1s - loss: 0.0069 - val_loss: 0.0080
Epoch 29/50
 - 1s - loss: 0.0073 - val_loss: 0.0119
Epoch 30/50
 - 1s - loss: 0.0064 - val_loss: 0.0032
Epoch 31/50
 - 1s - loss: 0.0061 - val_loss: 0.0058
Epoch 32/50
 - 1s - loss: 0.0083 - val_loss: 0.0084
Epoch 33/50
 - 1s - loss: 0.0083 - val_loss: 0.0024
Epoch 34/50
 - 1s - loss: 0.0056 - val_loss: 0.0069
Epoch 35/50
 - 1s - loss: 0.0075 - val_loss: 0.0071
Epoch 36/50
 - 1s - loss: 0.0072 - val_loss: 0.0042
Epoch 37/50
 - 1s - loss: 0.0095 - val_loss: 0.0039
Epoch 38/50
 - 1s - loss: 0.0062 - val_loss: 0.0245
Epoch 39/50
 - 1s - loss: 0.0107 - val_loss: 0.0052
Epoch 40/50
 - 1s - loss: 0.0089 - val_loss: 0.0044
Epoch 41/50
 - 1s - loss: 0.0094 - val_loss: 0.0108
Epoch 42/50
 - 1s - loss: 0.0091 - val_loss: 0.0068
Epoch 43/50
 - 1s - loss: 0.0091 - val_loss: 0.0061
Epoch 44/50
 - 1s - loss: 0.0062 - val_loss: 0.0037
Epoch 45/50
 - 1s - loss: 0.0068 - val_loss: 0.0128
Epoch 46/50
 - 1s - loss: 0.0074 - val_loss: 0.0035
Epoch 47/50
 - 1s - loss: 0.0041 - val_loss: 0.0062
Epoch 48/50
 - 1s - loss: 0.0062 - val_loss: 0.0092
Epoch 49/50
 - 1s - loss: 0.0084 - val_loss: 0.0049
Epoch 50/50
 - 1s - loss: 0.0064 - val_loss: 0.0047
```

Figure 8.2 A simple RNN model in Python (Continued).



In [11]:
```
yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], test_X.shape[2]))
inv_yhat = concatenate((yhat, test_X[:, 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
print('Test RMSE: %.3f' % rmse)
```

```
Test RMSE: 0.261
```

In [12]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_1 (SimpleRNN)     (None, 50)                3050
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 51
=================================================================
Total params: 3,101
Trainable params: 3,101
Non-trainable params: 0
_________________________________________________________________
```

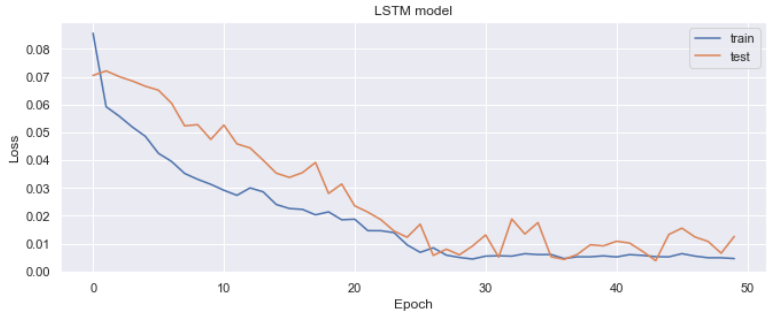
Figure 8.3 A simple RNN model in Python (Continued).

```
In [14]: model = Sequential()
         model.add(LSTM(50, input_shape=(train_X.shape[1], train_X.shape[2])))
         model.add(Dense(1))

In [15]: model.compile(loss='mae', optimizer='adam')

In [16]: history = model.fit(train_X, train_y, epochs=50, batch_size=100,
                             validation_data=(test_X, test_y), verbose=2, shuffle=False)
         Train on 8760 samples, validate on 82766 samples
         Epoch 1/50
          - 3s - loss: 0.0857 - val_loss: 0.0705
         Epoch 2/50
          - 2s - loss: 0.0593 - val_loss: 0.0721
         Epoch 3/50
          - 2s - loss: 0.0559 - val_loss: 0.0701
         Epoch 4/50
          - 2s - loss: 0.0520 - val_loss: 0.0685
         Epoch 5/50
          - 1s - loss: 0.0486 - val_loss: 0.0666
         Epoch 6/50
          - 1s - loss: 0.0425 - val_loss: 0.0652
         Epoch 7/50
          - 1s - loss: 0.0396 - val_loss: 0.0605
         Epoch 8/50
          - 1s - loss: 0.0353 - val_loss: 0.0524
         Epoch 9/50
          - 2s - loss: 0.0331 - val_loss: 0.0528
         Epoch 10/50
          - 2s - loss: 0.0313 - val_loss: 0.0475
         Epoch 11/50
          - 2s - loss: 0.0292 - val_loss: 0.0526
         Epoch 12/50
          - 2s - loss: 0.0274 - val_loss: 0.0459
         Epoch 13/50
          - 1s - loss: 0.0300 - val_loss: 0.0444
         Epoch 14/50
          - 1s - loss: 0.0286 - val_loss: 0.0400
         Epoch 15/50
          - 1s - loss: 0.0241 - val_loss: 0.0354
         Epoch 16/50
          - 2s - loss: 0.0226 - val_loss: 0.0338
         Epoch 17/50
          - 2s - loss: 0.0223 - val_loss: 0.0355
         Epoch 18/50
          - 2s - loss: 0.0204 - val_loss: 0.0392
         Epoch 19/50
          - 2s - loss: 0.0214 - val_loss: 0.0281
         Epoch 20/50
          - 2s - loss: 0.0186 - val_loss: 0.0314
         Epoch 21/50
          - 2s - loss: 0.0188 - val_loss: 0.0236
         Epoch 22/50
          - 2s - loss: 0.0147 - val_loss: 0.0213
         Epoch 23/50
          - 1s - loss: 0.0146 - val_loss: 0.0186
```

Figure 8.4 An LSTM model in Python.

```
Epoch 24/50
 - 1s - loss: 0.0139 - val_loss: 0.0145
Epoch 25/50
 - 1s - loss: 0.0096 - val_loss: 0.0123
Epoch 26/50
 - 1s - loss: 0.0069 - val_loss: 0.0170
Epoch 27/50
 - 2s - loss: 0.0085 - val_loss: 0.0058
Epoch 28/50
 - 1s - loss: 0.0059 - val_loss: 0.0080
Epoch 29/50
 - 1s - loss: 0.0050 - val_loss: 0.0060
Epoch 30/50
 - 1s - loss: 0.0045 - val_loss: 0.0091
Epoch 31/50
 - 1s - loss: 0.0055 - val_loss: 0.0131
Epoch 32/50
 - 1s - loss: 0.0056 - val_loss: 0.0051
Epoch 33/50
 - 1s - loss: 0.0055 - val_loss: 0.0189
Epoch 34/50
 - 1s - loss: 0.0064 - val_loss: 0.0135
Epoch 35/50
 - 2s - loss: 0.0061 - val_loss: 0.0176
Epoch 36/50
 - 1s - loss: 0.0061 - val_loss: 0.0053
Epoch 37/50
 - 1s - loss: 0.0046 - val_loss: 0.0043
Epoch 38/50
 - 1s - loss: 0.0053 - val_loss: 0.0061
Epoch 39/50
 - 2s - loss: 0.0053 - val_loss: 0.0096
Epoch 40/50
 - 2s - loss: 0.0056 - val_loss: 0.0092
Epoch 41/50
 - 2s - loss: 0.0052 - val_loss: 0.0108
Epoch 42/50
 - 2s - loss: 0.0060 - val_loss: 0.0102
Epoch 43/50
 - 1s - loss: 0.0058 - val_loss: 0.0072
Epoch 44/50
 - 1s - loss: 0.0053 - val_loss: 0.0039
Epoch 45/50
 - 1s - loss: 0.0052 - val_loss: 0.0133
Epoch 46/50
 - 2s - loss: 0.0064 - val_loss: 0.0156
Epoch 47/50
 - 1s - loss: 0.0055 - val_loss: 0.0124
Epoch 48/50
 - 1s - loss: 0.0049 - val_loss: 0.0108
Epoch 49/50
 - 1s - loss: 0.0049 - val_loss: 0.0066
Epoch 50/50
 - 1s - loss: 0.0046 - val_loss: 0.0126
```

Figure 8.5 An LSTM model in Python (Continued).




```
In [18]: # make a prediction
         yhat = model.predict(test_X)
         test_X = test_X.reshape((test_X.shape[0], test_X.shape[2]))

In [19]: inv_yhat = concatenate((yhat, test_X[:, 1:]), axis=1)
         inv_yhat = scaler.inverse_transform(inv_yhat)
         inv_yhat = inv_yhat[:,0]

In [20]: inv_yhat

Out[20]: array([3.272833 , 3.2825136, 3.2821205, ..., 5.4245396, 5.424094 ,
                5.4240484], dtype=float32)

In [21]: test_y = test_y.reshape((len(test_y), 1))
         inv_y = concatenate((test_y, test_X[:, 1:]), axis=1)
         inv_y = scaler.inverse_transform(inv_y)
         inv_y = inv_y[:,0]

In [22]: rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
         print('Test RMSE: %.3f' % rmse)

         Test RMSE: 0.403

In [23]: model.summary()

         Model: "sequential_1"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         lstm_1 (LSTM)                (None, 50)                12200
         _________________________________________________________________
         dense_1 (Dense)              (None, 1)                 51
         =================================================================
         Total params: 12,251
         Trainable params: 12,251
         Non-trainable params: 0
```

Figure 8.6 An LSTM model in Python (Continued).