CS230

A model for the prediction of influent water flows from wastewater treatment plants

Abhishek Krishnan
Civil & Environmental Engineering
abhikr@stanford.edu

Jorge Luis Meraz
Civil & Environmental Engineering
jmeraz@stanford.edu

Abstract

In this project we explore three distinct methods to predict wastewater treatment plant influent flows. Using our time series data, we develop two LSTM model architectures that use climate data to predict future water flows at varying time scales between 1 and 1000 hours. In addition, we develop a gradient boosting technique to compare against the performance of our deep learning models. We use the root mean squared error (RMSE) as the metric that our models attempt to minimize, in addition to using it as an overall measure of each model's performance. Our LSTM models each predict influent flows on an hourly scale and perform quite well, relative to the gradient boosting technique, having the lowest overall RMSE scores. The two distinct LSTMs differ in their architecture (e.g., model layers, hyperparameter tuning, predictive features), highlighting the various learning abilities of LSTMs in recognizing patterns in data.

1. Introduction

Increasing global populations have led to greater need and use for critical natural resources such as water. As water supplies decrease, replenishing depleted sources increasingly relies on wastewater treatment plants (WWTPs). WWTPs ensure that water is sufficiently treated before it is released back into the environment, decreasing the risks of adverse environmental pollution. Effectively treating water to adequate standards relies heavily on understanding how much water will be flowing through the plant during a given day. Being able to forecast influent water flows can optimize operational characteristics of the treatment plant to save time, energy, and capital. Here, we develop a series of simple regressive and deep learning approaches that utilize climate data to predict influent flows at time scales between 1 and 1000 hours. Specifically, we develop a gradient boosting model (XGBoost) and two distinct LSTM frameworks that predict influent flows using climate data (e.g., temperature, precipitation, humidity). The LSTM models that this project utilizes were initially developed to predict energy usage and traffic volume. We train these models on multiple weather parameters and use it to predict the influent flow. After initial model evaluation, the models are modified in order to better fit the dataset that we use by changing the hyperparameters, including the model architecture itself. Using these models we are able to predict influent flow at a much smaller temporal resolution of the forecast: no other existing model forecasts influent flow at an hourly scale. Having the ability to forecast far into the future is crucial because various processes carried out in wastewater treatment plants as well as decision making relies on being able to know the value of influent flow in advance.

2. Related work

Traditional approaches to forecast influent water flows use a combination of physical and autoregressive models [1], [2]. While these have proven useful, they are unable to capture the complex nature of aging infrastructure or climate impacts [1], [2]. To gain better understanding of these complex relationships, data-driven models have been increasingly applied to forecasting water influent flows. Previous studies have implemented deep learning architectures, particularly recurrent neural networks (RNNs), to forecast key WWTP characteristics such as influent temperature, influent biochemical oxygen demand, and influent flow [2]–[4]. Fernandez et al. 2009 implemented a FNN using two input variables (day of the week and average daily flow-rate), and was able to forecast flows up to one month with average errors below 10%. Oliveira et al. 2020 used LSTMs and CNNs to understand the relationship between influent flow and various climate features, such as humidity and precipitation [3]. In this work, LSTMs and 1-D CNNs were compared relative to their ability to forecast influent flows on a daily scale. The best candidate model was the LSTM, which had the ability to forecast flows three days out with an overall error of 200 [m3]. Inspired by the work of Oliveira et al. 2020, we focus our efforts on building a deep learning LSTM model to predict influent water flows using climate data. For our model, we use three input climate features – temperature, precipitation, and humidity – to predict influent flows. Our end goal is to get a deeper appreciation of the predictive power of RNNs by comparing the performance of various LSTM architectures to that of multivariate techniques.

3. Dataset and Features

We used two time-series datasets for this project. Wastewater treatment plant data was obtained from Silicon Valley Clean Water, a wastewater treatment plant in Redwood City, California that serves a number of cities in San Mateo County. The full dataset contains 33 features collected in time intervals of 15 mins. From this feature set we isolated influent wastewater flows, reported in millions of gallons, aggregating them into their respective hourly sums. Climate data was obtained from Visual Crossing API. This dataset contains 22 distinct weather conditions collected on hourly intervals. In this dataset we isolated temperature, humidity, and precipitation. Temperature, humidity, and precipitation are reported in celsius, percentage, and millimeters, respectively. Both time series datasets were structured similarly and merged into a single dataframe for subsequent analyses. The following table and figure show an overview of the time series characteristics of our data, as well as summarizes basic statistics of each dataset feature.

Table 1. Baseline statistics of feature dataset

Feature	Mean	Std. Dev	Min/Max
Influent Flow (MGD)	53	24.5	0/280
Temperature (C)	15	4.4	0/39.5
Precipitation (mm)	0.04	0.3	0/13.4
Humidity (%)	69	15	0/134

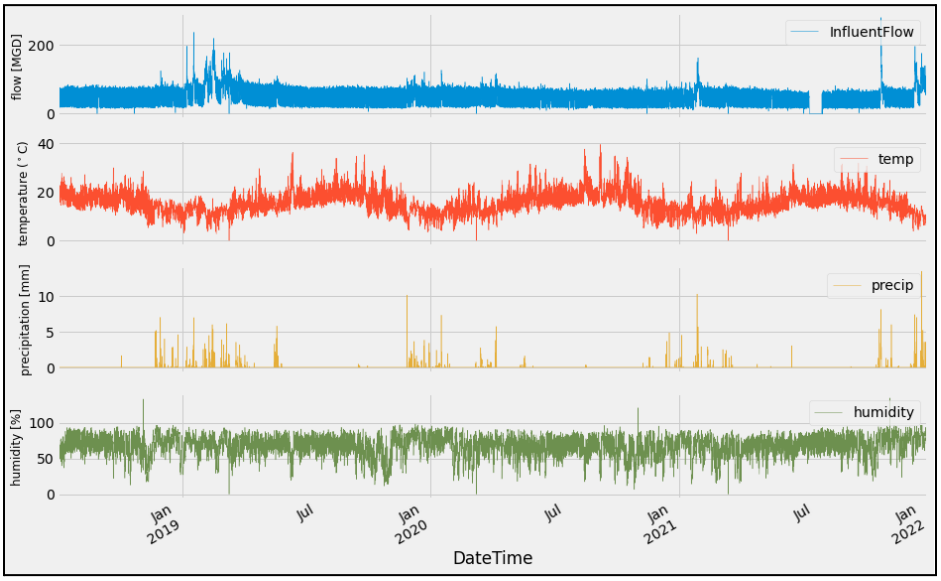

Figure 1. Time Series dataset

The dataset was fairly comprehensive, but also has some limitations that are likely to have affected the performance of the model used. For one, there were small but notable clusters of continuous data points with negative or null values for the influent flow due to recording errors. For consistency's sake, those values were replaced with zeroes, which could impact the accuracy of the training of the model and the validation. Additionally, the number of data points was around 30,000, which is a large enough sample to capture the diurnal nature of wastewater flow, with it peaking in the morning and in the evening. However, for training purposes, with the values of input features such as temperature and precipitation likely being the same on multiple days, this is another potential source of error. This is particularly pronounced for precipitation, where there are several days with zero values.

4. Methods

Baseline

A LSTM model architecture was implemented as the baseline method to predict influent wastewater flows. To carry out this baseline method, the dataset, including all features, was transformed into a multidimensional matrix array with a time lag=1 for each feature. The new, transformed dataset includes 4 time lag variables and 1 independent variable at time=t. For training, all array values are initially scaled to values between 0 and

1. For the training data, 1 full year of data is used, while the remaining data is allocated to the test set. For the baseline model, a sequential LSTM model is constructed with 3 layers – a LSTM layer with 100 neurons, a dropout layer of 20%, and a dense layer. The loss function used for our model was the mean squared error (MSE), with Adam chosen as the model specific optimizer. MSE and Adam were used as they seem to be the standard for time series specific models using an LSTM architecture [5], [6]. Model results, including loss function performance and predictive capability, can be found in the subsequent Results section. In addition to the LSTM, we evaluate the predictive capability of a gradient boosting technique, XGBoost. A brief discussion and model results from the XGBoost analysis can be found in the Appendix.

Novel method and hyperparameter tuning
The novel LSTM model used is a fairly simple LSTM based upon a previously implemented model for predicting metro traffic [7]. The model architecture consists of an LSTM layer, followed by dropout, and ending with a dense output layer. The model aims to forecast the influent flow 1000 hours in advance, as compared to the baseline model which forecasts the flow 1 hour ahead of time. However, unlike the baseline model, the influent flow itself is not used as a feature. Table 2 below contains the default hyperparameters used while training the model. The hyperparameters chosen were similar to those chosen for the baseline model, for the same reasons mentioned in the previous section.

Table 2. Hyperparameter overview

Hyperparameter	Value
Number of neurons	100
Batch size	128
Dropout probability	128
Loss function	Mean squared error
Optimizer	Adam
Number of epochs	150
Steps per epoch	100

5. Results and Discussion

Overall model performance
In Table 3 we highlight each model's performance as evaluated by the RMSE. The baseline LSTM model outperforms the XGBoost and higher order LSTM models by ~60% and ~40%, respectively.

Table 3. Model performance overview

Model	RMSE
Baseline LSTM (Best)	9.70
XGBoost (see Appendix)	23.95
Higher Order LSTM (pre-tuning)	16.62

The higher performance of the baseline LSTM model can be attributed to the use of a time lagged influent flow parameter as a feature, in addition to the time lagged climate variables. The XGBoost and higher order LSTM model do not make use of time lagged or influent flow variables as features. As a result of this, the models tend to exhibit lower performance, but may be more generalizable. Moreover, given larger climate datasets, RMSE for XGBoost and the higher order LSTM have the potential to converge to a value closer to that of the baseline model.

Hyperparameter tuning
The results of the hyperparameter tuning exercise conducted on the higher-order LSTM are shown below in

Table 4. The best configuration has been highlighted. From the tuning procedure carried out, the important takeaways were that the model performed better with lower dropout probability, higher batch size, with mean absolute error as the loss function, and with the optimizer being Adam.

Table 4. Results of hyperparameter tuning

Hyperparameter values					Metrics		
Number of neurons	**Dropout probability**	**Batch size**	**Loss function**	**Optimizer**	**Mean Absolute Error (MAE)**	**Root Mean Squared Error (RMSE)**	**R^2**
100	0.2	128	MSE	Adam	13.65	16.62	0.0733
50	0.2	128	MSE	Adam	18.63	21.96	-0.638
200	0.2	128	MSE	Adam	20.936	25.62	-1.227
100	0.5	128	MSE	Adam	16.74	19.90	-0.344
100	0.2	128	MAE	Adam	13.58	16.59	0.065
100	**0.2**	**256**	**MAE**	**Adam**	**12.84**	**15.49**	**0.186**
100	0.2	256	MAE	RMSprop	13.32	15.92	0.137
100	0.2	256	MAE	Adadelta	23.35	28.92	-1.840
100	0.2	256	MAE	SGD	15.09	20.10	-0.372

LSTM models

The following figures highlight the performance of the baseline and higher order LSTM models. Figures 3-6 illustrate the predictive capability of forecasting future influent water flows at the 1 hour (baseline) and 1000 hour (higher order LSTM) marks. Essentially, the goal is to see how far in advance we could predict wastewater flows, given only climate data features, and with minimal loss of model performance. The baseline model performs quite well, which is expected, given that it uses a time lagged variable of influent flow as a predictive feature. While this showcases the predictive power of LSTMs, we are interested in modifying our LSTM model architecture to predict beyond the 1 hour mark and with features other than influent flow.

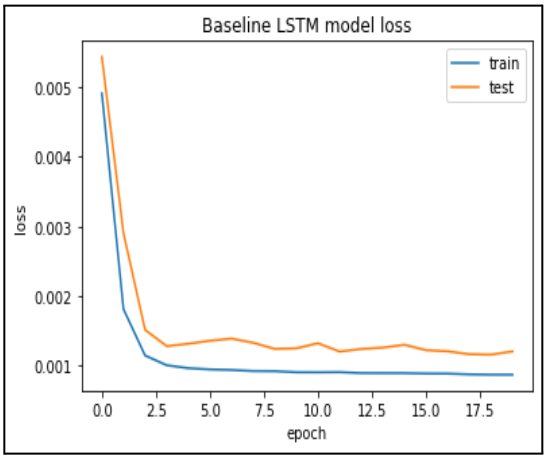

Figure 2. Baseline LSTM loss

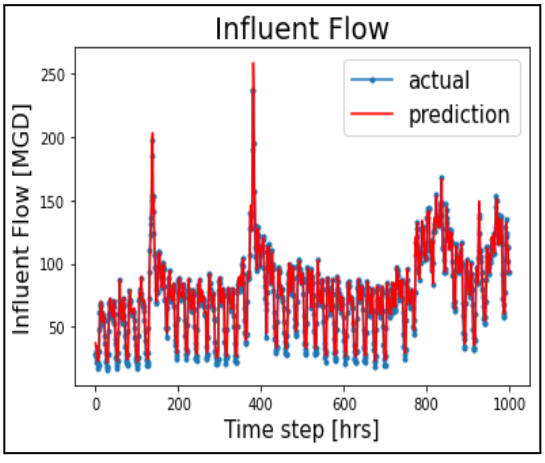

Figure 3. Baseline LSTM predictive output (time lag period=1)

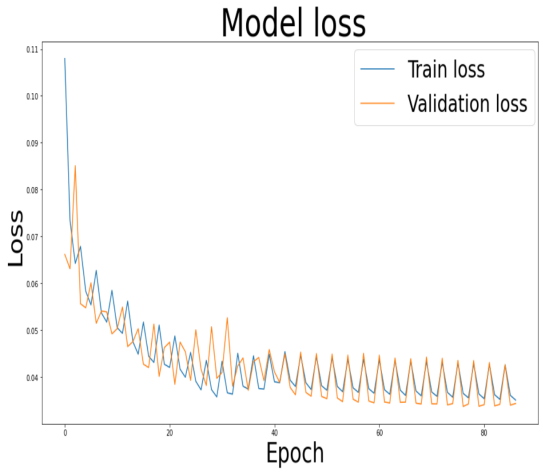

Figure 4. Tuned novel LSTM loss

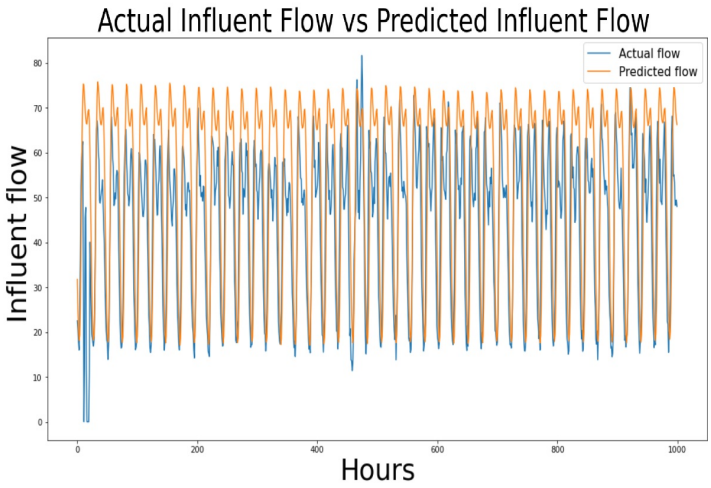

Figure 5. Tuned novel LSTM predictive output (horizon = 1000 hours)

The goal of the novel LSTM was much more ambitious than that of the baseline LSTM, but it still performed relatively well. The training and validation loss were relatively well aligned. The diurnal nature of the wastewater flow (peaks at two times in the day) was captured pretty well, and the model was somewhat sensitive to sharp changes in weather conditions. However, even with layer normalization and dropout regularization implemented, it was clear that there is a degree of overfitting occurring. Tuning was able to improve the model's performance, but the overshooting of the values persisted. This was especially noticeable when the validation set had high variability.

6. Conclusion/Future Work

The performance of the higher order LSTM model was relatively good when the test set influent flow values were relatively similar to the training data, and the model was able to capture the diurnal nature of the influent flow time series. However, the model's performance was hampered by the overfitting to the small dataset. To reduce it, future iterations of the model would benefit from use of different kinds of regularization, such as L1 or L2 regularization, to supplement the dropout. Procuring a larger dataset with less variability and less anomalies would also help.

We would also attempt to carry out hyperparameter tuning using a more systematic algorithm, such as grid or randomized search. Moreover, rather than representing precipitation as a discrete variable, it can be represented as a masked variable indicating whether precipitation was present or not. This masking could potentially increase the generalizability of our model, in addition to increasing model performance. Lastly, the possibility of using other time-series input variables apart from the weather data would be considered, with the aim of improving performance even further.

7. Contributions

AK contributed to the project code and write-up. JLM contributed to the project code and write-up.

References

[1] Q. Zhang, Z. Li, S. Snowling, A. Siam, and W. El-Dakhakhni, “Predictive models for wastewater flow forecasting based on time series analysis and artificial neural network,” Water Sci. Technol., vol. 80, no. 2, pp. 243–253, Jul. 2019.

[2] T. Cheng, F. Harrou, F. Kadri, Y. Sun, and T. Leiknes, “Forecasting of Wastewater Treatment Plant Key Features Using Deep Learning-Based Models: A Case Study,” IEEE Access, vol. 8. pp. 184475–184485, 2020. doi: 10.1109/access.2020.3030820.

[3] P. Oliveira, B. Fernandes, F. Aguiar, M. A. Pereira, C. Analide, and P. Novais, “A Deep Learning Approach to Forecast the Influent Flow in Wastewater Treatment Plants,” in Intelligent Data Engineering and Automated Learning – IDEAL 2020, 2020, pp. 362–373.

[4] F. J. Fernandez, A. Seco, J. Ferrer, and M. A. Rodrigo, “Use of neurofuzzy networks to improve wastewater flow-rate forecasting,” Environmental Modelling & Software, vol. 24, no. 6, pp. 686–693, Jun. 2009.

[5] B. B. Sahoo, R. Jha, A. Singh, and D. Kumar, “Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting,” Acta Geophys., vol. 67, no. 5, pp. 1471–1481, Oct. 2019.

[6] Z. Chang, Y. Zhang, and W. Chen, “Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting,” in 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Nov. 2018, pp. 245–248.

[7] B. V. Vishwas and A. Patel, “Hands on time series analysis with Python,” GitHub, 2020. https://github.com/Apress/hands-on-time-series-analylsis-python#apress-source-code (accessed 2022).

Appendix

The XGBoost method uses our merged data, without any lags, and is built using ~2 years of data for our training dataset. Similar to the baseline neural network model, RMSE is used as our metric to evaluate how well the model performs (See Table 3 in main text). XGBoost can be implemented with a number of hyperparameters, but for the purposes of this work, we are primarily interested in evaluating how well the model performs relative to the LSTM networks, without intensive hyperparameter tuning. The XGBoost model is built using 5000 sequential trees, as this was found to be the number of estimators that minimized the loss function.

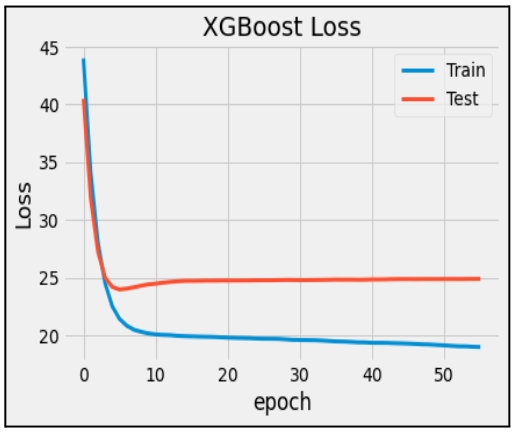

Figure 1. XGBoost loss

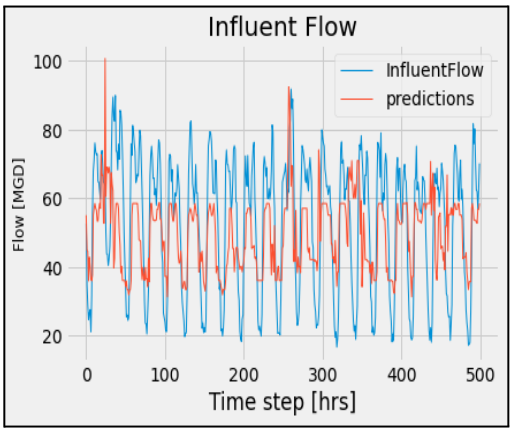

Figure 2. XGBoost predictive output