Water Turbidity prediction and Comparison of machine learning approaches, probabilistic and deep learning models, Lake Michigan, Chicago

Almog Klein

Brian Rikshpun

Afeka College of Engineering

June 25, 2022

1 Abstract

Smart city, a city full of data collected by different sensors. Data that has a huge potential to help answer important questions. A good example of a smart city is Chicago, and a good example of an important question is Lake Michigan's turbidity level prediction.

This is exactly what we are trying to predict in our research by using different machine learning (ML) and deep learning (DL) algorithms, as shown in [1][2][3]. We will compare two datasets, The first one is a dataset of water data collected by water sensors and the second one is the merging of the data collected by the water sensors and the data collected by weather sensors in Michigan lake, both datasets merged with the sensors location dataset. We will compare between different prediction and historical data intervals and how it effects on the quality of the models.

2 Introduction

The Chicago Park District maintains weather sensors at beaches along Chicago's Lake Michigan lakefront (collecting weather data) and inside the water (collecting water data). These sensors generally capture the indicated measurements hourly.

Image 1 - Left, different levels of turbidity. Right, lake map and pictures of the various sensors

The turbidity is caused by particles suspended or dissolved in water that scatter light, making the water appear cloudy or murky. High turbidity can significantly reduce the aesthetic quality of lakes and streams, having a harmful impact on recreation and tourism. It can increase the cost of water treatment for drinking and food processing can also harm fish and other aquatic life by reducing food supplies, degrading spawning beds and gill function.

Machine learning Models:

Logistic Regression - A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. Logistic models can also transform raw data to create features for other types of machine learning techniques. Logistic regression is one of the commonly used algorithms in machine learning for binary classification problems.

Decision Tree Classifier - Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions. Can perform both classification and regression tasks.

K-Neighbors Classifier - The k-nearest neighbors algorithm (k-NN) is a non-parametric supervised learning method. It is used for classification and regression. In both cases, the input consists of the k closest training examples in a data set. The output depends on whether k-NN is used for classification or regression. Classification, an object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. Regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

Random Forest Classifier -

Is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

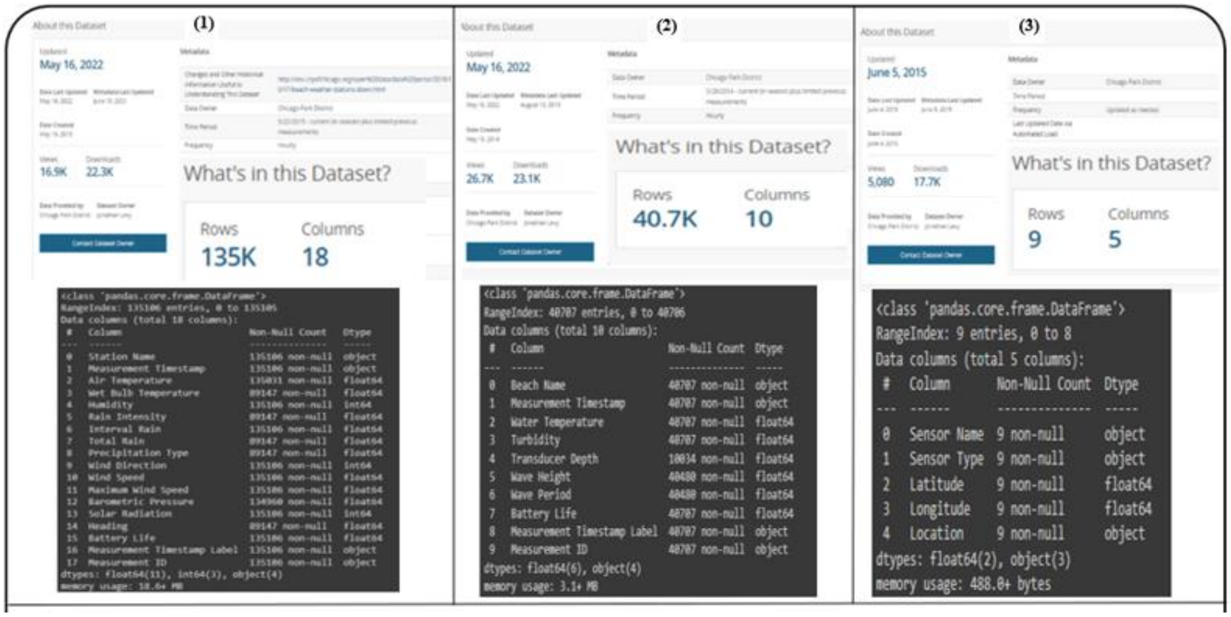
Deep learning Model:
LSTM – An artificial neural network, unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network can process not only single data points (such as images), but also entire sequences of data.

3 Data description

The data for this forecasting task was taken from the city of *Chicago's open data portal*. It contains city data, facts about your neighborhood, create maps and graphs about the city, and freely download or get an API for the desired data.

Many of these datasets are updated at least once a day, and many of them are updated several times a day.

Due to the above we have decided to use three databases related to the quality of resources around Lake Michigan and examine the prediction results which are obtained from the data separately and consolidated.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 135106 entries, 0 to 135105
Data columns (total 18 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   Station Name                 135106 non-null  object
 1   Measurement Timestamp        135106 non-null  object
 2   Air Temperature              135031 non-null  float64
 3   Wet Bulb Temperature         89147 non-null   float64
 4   Humidity                     135106 non-null  int64
 5   Rain Intensity               89147 non-null   float64
 6   Interval Rain                135106 non-null  float64
 7   Total Rain                   89147 non-null   float64
 8   Precipitation Type           89147 non-null   float64
 9   Wind Direction               135106 non-null  int64
 10  Wind Speed                   135106 non-null  float64
 11  Maximum Wind Speed           135106 non-null  float64
 12  Barometric Pressure          134960 non-null  float64
 13  Solar Radiation              135106 non-null  int64
 14  Heading                      89147 non-null   float64
 15  Battery Life                 135106 non-null  float64
 16  Measurement Timestamp Label  135106 non-null  object
 17  Measurement ID               135106 non-null  object
dtypes: float64(11), int64(3), object(4)
memory usage: 18.6+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40707 entries, 0 to 40706
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Beach Name                   40707 non-null  object
 1   Measurement Timestamp        40707 non-null  object
 2   Water Temperature            40707 non-null  float64
 3   Turbidity                    40707 non-null  float64
 4   Transducer Depth             10034 non-null  float64
 5   Wave Height                  40480 non-null  float64
 6   Wave Period                  40480 non-null  float64
 7   Battery Life                 40707 non-null  float64
 8   Measurement Timestamp Label  40707 non-null  object
 9   Measurement ID               40707 non-null  object
dtypes: float64(6), object(4)
memory usage: 3.1+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Sensor Name  9 non-null      object
 1   Sensor Type  9 non-null      object
 2   Latitude     9 non-null      float64
 3   Longitude    9 non-null      float64
 4   Location     9 non-null      object
dtypes: float64(2), object(3)
memory usage: 488.0+ bytes
```

Image 2 - (1) Weather, (2) Water, (3) Location datasets

4 Methodology of work

The first step we took in our journey is understanding the data, we made a deep EDA (explanatory data analysis) and cleaning process on the datasets. Interesting fact we found out was that Foster Weather Station misses few sensors and because of that 35% of the data in few columns was empty.

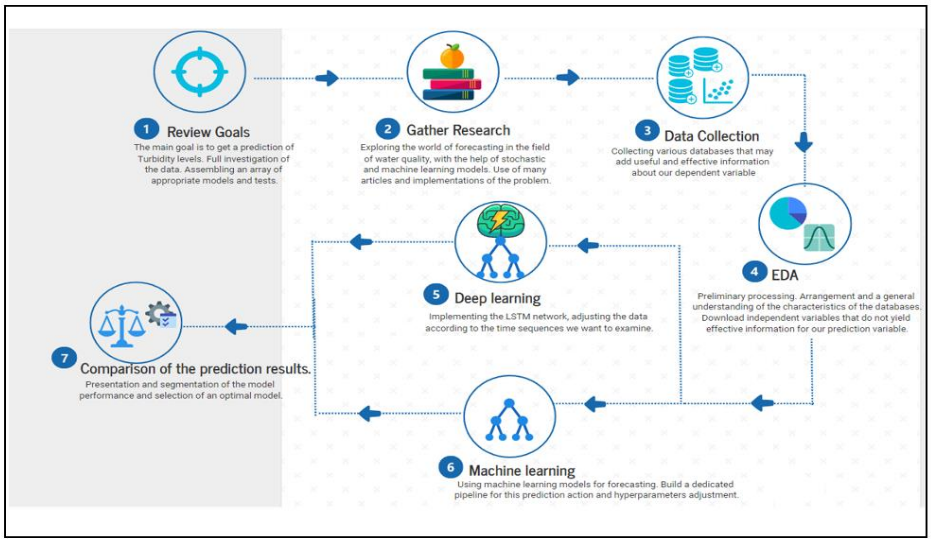

1 Review Goals
The main goal is to get a prediction of Turbidity levels. Full investigation of the data. Assembling an array of appropriate models and tests.
2 Gather Research
Exploring the world of forecasting in the field of water quality, with the help of stochastic and machine learning models. Use of many articles and implementations of the problem.
3 Data Collection
Collecting various databases that may add useful and effective information about our dependent variable
4 EDA
Preliminary processing. Arrangement and a general understanding of the characteristics of the databases. Download independent variables that do not yield effective information for our prediction variable.
5 Deep learning
Implementing the LSTM network, adjusting the data according to the time sequences we want to examine.
6 Machine learning
Using machine learning models for forecasting. Build a dedicated pipeline for this prediction action and hyperparameters adjustment.
7 Comparison of the prediction results.
Presentation and segmentation of the model performance and selection of an optimal model.

figure 3 - Comparison of model models according to the prediction time interval

We decided to delete the station from our water dataset. We also deal with outliers we found in different features and removed columns with low variance of data.

After the data cleaning we merged between the water and weather datasets and added a new column in each one called “class”, any observation with turbidity higher than 5 was classified as 1 (high turbidity value) and the rest as 0 (low turbidity value) for our classification task. A quick look at our class at each dataset yields the conclusion of the imbalanced classification task, our data is imbalanced, and we used the SMOTE [3] algorithm to balance our training data.

When we finished cleaning the data, we divided the data and balanced the classes in the train of each dataset. To try and predict the class of the turbidity level we used 4 different machine learning algorithms: Logistic regression, KNN classifier, Decision Tree classifier and Random Forest classifier. For each one we found the best hyper-parameters using gridsearchCV and compared between them by the AUC score.

We also implemented LSTM network – We chose to implement a standard network from an LSTM structure. We want to use our network to predict from sequences of days. For this purpose, we used a network structure called an encoder decoder.[4]

The encoding level, learn the relationship between the steps in the input sequence and develop an internal representation of these relationships. The encoder model can be implemented using one or more LSTM layers. This model produces a fixed-size vector representing the internal representation of the input sequence. The length of this fixed-sized vector is determined by the number of memory cells in this layer.

The decoder, must transform the learned internal representation of the input sequence into the correct output sequence. This model reads from the fixed sized output from the encoder model. As the network's output, a dense layer is used. By wrapping the dense layer in a TimeDistributed wrapper, the same weights can be used to output each time step in the output sequence.

5 Results

We tested the machine learning models predictions up to seven days ahead prediction, the results for each of them

	Without PCA		With PCA	
prediction ahead	accuracy	precision	accuracy	precision
1	80%	50%	71%	30%
2	74%	46%	73%	31%
3	69%	33%	74%	38%
4	83%	36%	79%	47%
5	79%	45%	67%	30%
6	74%	44%	67%	22%
7	69%	35%	69%	38%

Table 2 - Comparison of model models according to the prediction time interval

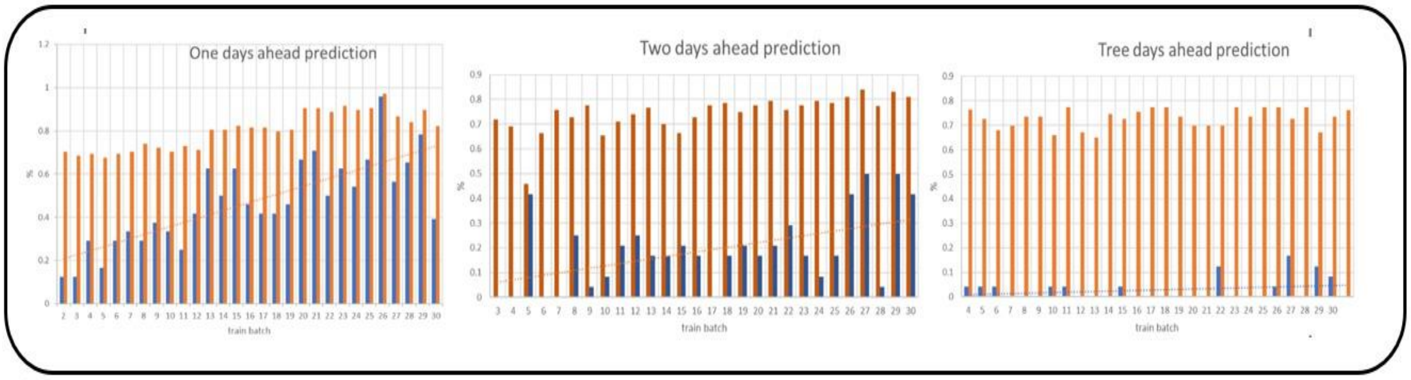


Figure 1 - Results obtained by the LSTM network (orange – accuracy, blue – precision)

The above table show the prediction results obtained from the LSTM network.

As can be seen, there is an increase in the precision trend, and we get 95% -97% precision - accuracy from learning sequences of 26 days, similar results shown at, learning on sequence of 24 days [5].

Days of prediction ahead	ML/DL	Dataset	accuracy	precision
1	DL	Water	97%	96%
2	DL	Water	84%	50%
3	ML	Water	74%	38%
4	ML	Water	83%	36%

Table 2 – Summarizing the results obtained by ML and DL

6 Conclusions

As the results indicate, adding the weather data to predict the turbidity was inefficient. The best models we found to predict the turbidity 1 – 4 days ahead didn't base on the water and weather dataset.

The PCA didn't yield better results as expected (decreasing the number of dimensions should help us get better classification results because we decrease the dimension curse impact).

We can see that the LSTM (DL) is more effective in predicting 1 - 2 days ahead the turbidity class with a sequence of 26 days input, and the ML has better results in predicting the 3 – 4 days ahead.

7 Reference

[1] – "Predicting water quality at Santa Monica Beach: Evaluation of five different models for public notification of unsafe swimming conditions", W. Thoe a,* , M. Gold b , A. Griesbach c , M. Grimmer c , M.L. Taggart c , A.B. Boehm, 2014.

[2] - "Monitoring Stream Water Quality: A Statistical Evaluation"

[3] - "SMOTE: Synthetic Minority Over-sampling Technique"

[4] - "Study on turbidity prediction method of reservoirs based on long short-term memory neural network"

[5] - "New double decomposition deep learning methods for river water level forecasting"