Andres Tapia

Capstone Project

Spring 2002

AI/ML model for correcting Temperature values and compensating for self-heating on Waggle/Sage nodes

Abstract

The main goal of this project is to design and build a machine learning model and to train the model to make hardware improvements on the temperature reading capabilities of field-deployed Waggle sensors. While the sensors measure temperature, humidity, and pressure from the environment, analysis from the sensors’ data indicated that the sensors’ values deviate from the actual values of the environment, provided by an Argonne meteorological tower in the area, as they are corrupted by the heat that is given off from the computing system accompanying the sensor (self-Heating). Based on this fact, the goal of this project is to correct the Temperature values using a machine learning model to accurately predict the correct temperature. Before the data is used, it is processed and cleaned using the Python library pandas. An array of sensor values (~7 parameters) is collected from a node every 15 minutes. The model is a linear regression model that uses this array of parameters as features, or characteristics used for analysis, to predict the correct temperature. The project is part of a larger initiative called Sage, which is a project funded by the National Science Foundation to design and build a new kind of national-scale reusable cyberinfrastructure to enable AI at the edge.

Introduction

Methods and Materials

The time-series data is collected from Waggle sensors that are placed around the country. The sensor data is then stored in a Sage database, which houses data ranging from temperature, humidity, pressure, and rainfall measurements. The node that was chosen for this project was node W039, which is located on-site at Argonne National Laboratory, Lemont, Illinois, and is near the meteorological tower that measures the accurate temperature from the environment (Figure 1). To access the data, query calls to the Sage API are made and they return the desired measurements, or in this case, the temperatures measured by the node’s temperature sensors. Additional data from the meteorological tower is collected using the same query method but from a different source. Prior to analyzing the data, it is processed and cleaned using the Python library pandas so that the relevant data is returned and visible for analysis. Once the data is ready, it is loaded from a .csv file and put into a panda DataFrame where features, such as the hour of day and month, are added into the dataset to help train the machine learning model to correctly predict the temperature that is accurately measured by the meteorological tower. Once that step is complete, the data is split into training and test sets and the temperature from the tower is separated so the model knows to predict that variable. Utilizing the Python libraries tensorflow and keras, the data is used to train a linear regression model consisting of multiple inputs to predict the temperature measured from the tower. Although the model predicted the values correctly, an additional model was made to reduce error. This model is a deep neural Diagram, map

Description automatically generatednetwork, in which there are more layers that help predict the temperature.

Figure 1: Location of node W039. Site: Argonne National Laboratory, Lemont, IL

Results and Discussion

Before building the ML model, analysis was done on the dataset that has both the temperature values from the sensor and the meteorological tower. The histogram on Figure 2 shows that there is a large difference between the two temperature readings, thus signifying that the temperature sensor in the node is returning inaccurate values when compared to the meteorological tower. To alleviate this problem, the potential solution is to build a ML model that correctly predicts the temperature measured by the ATMOS tower. Table 1 is a pandas DataFrame that shows the input features such as the hour of day and month, the temperature read from the ambient and system temperature sensors from the node, as well as the accurate temperature read from the tower. The temperature column ‘temp\_atmos\_10’ is what the ML model correctly predicts using the features input from the node sensor. To build the ML model, the data was split into training and testing datasets. The training data trains the ML model to predict the accurate temperature and the test data is used after the model predicts the values to see how accurate it was in predicting the temperature. The ML model in question was a linear regression model that used the previously mentioned input features to predict the temperature based on those features. After the features were normalized, the Python libraries tensorflow and keras were used to fit the model with the training data and to then predict the temperature. Figure 3 shows the result of the prediction, where the straight line represents the model’s temperature prediction on the test data, and the scattered dots represent the actual temperature from the test data. While it predicts rather reliably, another ML model was also looked at to potentially reduce the prediction error that came with the linear regression model. This new model, the deep neural network (DNN), operates similarly to the linear regression model, only there are more layers at work in this model. The training data was fit into the DNN model to train and predict the correct temperature.

Chart, histogram

Description automatically generated

Figure 2: Difference between temperatures measured by ambient sensor and meteorological tower.

Table 1: Sample of input data from the node and tower, with added features for ML model.

Table

Description automatically generated

Chart, scatter chart

Description automatically generated

Figure 3: Difference between predicted and actual temperature from meteorological tower applied by the linear model.

The results of using this model yielded a lower error rate than the linear regression model, as shown in Figure 4 and Table 2, according to the Mean Absolute Error loss function. Figure 5 shows the distribution of error from the predicted values when compared to the test values from the initial dataset. The distribution shows a small margin of error between +1 and -1, meaning that there is very little error from the predictions. Therefore, from these results, the deep neural network ML model is best suited to predict the temperature measured by the meteorological tower on-site at Argonne National Laboratory.

Chart, scatter chart

Description automatically generated

Figure 4: Difference between predicted and actual temperature from meteorological tower applied by the deep neural network model.

Table 2: Test results showing prediction error between the two different ML models.

A picture containing text

Description automatically generated

Chart, histogram

Description automatically generated

Figure 5: Distribution of error from predicted temperature compared with actual temperature.

Conclusions

References

Acknowledgements

This work has been conducted under the mentorship and guidance of Dr. Rajesh Sankaran and Mr. Scott Ehling from Argonne National Laboratory, Mathematics and Computer Science Division and Prof. Elena Zheleva from the University of Illinois at Chicago, Department of Computer Science.