Enhancing Prediction of Sensor Data Using Random Forest Regressor: A Comprehensive Regression Analysis

MUDRA VERMA[1]
Faculty of Computer Science,
Dalhousie University,
6050 University Ave, Halifax,
Nova Scotia, Canada, B3H 4R2
mudraverma@dal.ca

DHRUV NRUPESHBHAI PATEL[2]
Faculty of Computer Science,
Dalhousie University,
6050 University Ave, Halifax,
Nova Scotia, Canada, B3H 4R2
dhruvpatel@dal.ca

Abstract: The knowledge of data is one of the most crucial factors in developing a machine learning model. A clear approach to the problem and ways to achieve desired results are narrowed down with clarity in the data. However, in this paper we propose our research on creating a sensor data prediction model with unlabeled time series data. The dataset consisted of 4 years of weekly training data of 103 sensors with failure scores for each corresponding week as training data. Furthermore, sensor data for a year is provided and our objective is to predict the corresponding failure scores.

Keywords: Data Prediction, Machine Learning, Cross Validation, Time Series data, Cross Validation, Random Forest Regressor

I. INTRODUCTION

Time series data are values which are recorded and observed over a period. It is arranged in chronological order and depends on the index [7]. In our problem we are tasked to deal with 4 years of training data of 103 sensors to predict the outcome of the following year. There exists a high degree of correlation between a few of the sensor data which makes prediction difficult. It is also essential to capture this correlation during prediction. One of the major difficulties in training the model is the absence of actual results of test data. So, the model is trained using only the training set and the test set is used only for evaluating the final performance of the model. The goal of the model is to predict a numerical output variable based on input data. Hence, this is a regression problem and we decide to use a regression algorithm.

Random Forest Regressor - Random Forest is an ensemble learning machine learning model that uses multiple decision trees to train the model. The output of the model is better as it uses multiple perfectly trained decision trees on different sample data to perform predictive analysis. It uses averaging for predictions and controls over-fitting. It is also effective in handling non-linear relationships between the independent variables and the dependent variable. Random Forest has a disposal of features to choose from and each decision tree is independent from each other [2]. However, the training time of Random Forest Regressor is very long. Our model took approximately 25 to 30 seconds to compile which might be even higher for large datasets. This might also be because we have used Repeated K-Fold Cross Validation.

We used Repeated K-fold cross-validation instead of Hold-out cross validation or K-Fold cross validation to split the data for training and validation. While K-fold cross-validation procedure is a standard method for estimating the performance of algorithm or configuration on a dataset, it may result in a noisy estimate of model performance [6]. Repeated k-fold cross-validation, on the other hand, provides a way to improve the estimated performance of a machine learning model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs. This is expected to be a more accurate estimate of the true unknown underlying mean performance of the model on the dataset, as calculated using the standard error [6]. We evaluated the accuracy of the model by calculating RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) of the model on each fold. The mean and standard deviation of these scores is reported as the final performance metrics. After training the data, prediction is made on the test set which is the final target variable in the test data.

II. DATASET AND FEATURES

The dataset provided consists of some correlation with its non-linear data [1]. We initially considered feature selection using F-score to identify the most important features that contribute the most to the prediction accuracy of the model. We used the top features identified using SelectKBest to train our model. However, we found that it led to loss of information and negatively impacted the model performance.

III. METHODOLOGY

A. Training the model

The method used to train the model in the above code is Repeated K-fold cross-validation. The data is divided into K equally sized folds where one-fold is used for validation and the remaining K-1 folds are used for training the model [8]. The process is repeated for a specified number of times (repeats) to get more reliable estimates of the model performance. In our approach we initialized the Repeated K-Fold cross-validation object with 10 splits and 10 repeats, resulting in a total of 100 iterations. We run a loop for the number of folds and at the end of the loop predict the result on the test data. This approach allows for a more robust evaluation of the model's performance, as it avoids overfitting and provides an estimate of the model's generalization performance on unseen data [8]. It also allows for a better understanding of the variability of the model's performance across different subsets of the training data [8].

B. Model Predictions

Prediction of the target variable is achieved using the Random Forest Regressor model. Random Forest combines multiple decision trees to create a more robust and accurate model. Since the training data is non-linear, we implemented the Random Forest Regressor model which can capture complex interactions between features. It is also robust against overfitting, since each tree is trained on a different subset of the data and features, and the final prediction is an average of the predictions of all the trees. The model is trained on the training data on each fold and used to predict the validation data for each fold. After the loop has ended predictions are made on the test set using the trained model.

C. Evaluation Metrics

We used RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) to evaluate the model in predicting the target variable on the validation data. RMSE is a commonly used metric for regression problems, and it measures the average deviation of the predicted values from the actual values. The lower the RMSE value, the better the model's performance. MAE measures the average absolute difference between the predicted values and the actual values. By calculating both RMSE and MAE we can provide a comprehensive measure of the model's accuracy on the validation data.

In our implementation, RMSE and MAE are calculated for each fold resulting in a total of 100 values each. The mean and standard deviation of these scores is reported as the final performance metrics. Our model calculated the following scores - Mean RMSE: 17.78 (std: 2.62) Mean MAE: 13.96 (std: 2.22).

IV. COMPARISONS

A. Multi-Layer Perceptron Regressor

The MLP Regressor is a type of supervised learning model that uses backpropagation to optimize the weights of the network based on the error between the predicted output and the actual output [4]. It contains multiple hidden layers, and each layer contains a set of neurons that are connected by weights. Most importantly, it can learn complex non-linear relationships between input and output variables.

The other model we implemented for evaluation is Multi-Layer Perceptron Regressor with Repeated K-fold cross-validation. MLP is popular for time series data because of its ability to model nonlinear relationships. The Repeated K-Fold object is created with the 10 splits and 10 repeats. Then, the code iterates over the folds of the cross-validation which is 100. It trains the model on the training data by making predictions on the validation data. The RMSE and MAE are calculated for each fold. Finally, the mean and standard deviation of the RMSE and MAE across all folds and repeats are

calculated and printed to evaluate the overall performance of the model.

The mean and standard deviation of the RMSE and MAE across all folds and repeats are calculated to evaluate the overall performance of the model. The model calculated the following scores - Mean RMSE: 23.58 (std: 3.15) Mean MAE: 19.21 (std: 2.73).

Hence, lower the RMSE and MAE better is the accuracy of the model. Due to which Random Forest Regressor was used to make the final predictions.

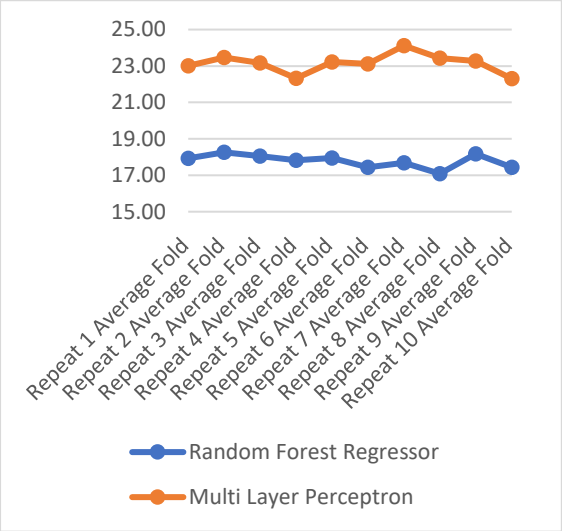

Figure 1: Average RMSE values for MLP & Random Forest

V. CONCLUSION

This study researched and implemented Random Forest Regressor and Multi-Layer Perceptron Regressor models for predicting results on non-linear time series data. The challenge faced was in validating the model due to absence of test data scores. The models used Repeated K-fold cross-validation in splitting the data to train and validate the model. The accuracy was calculated on the RMSE and MAE of the model. Since, lower the RMSE and MAE better is the measure of the model. Due to which Random Forest Regressor was used to make the final predictions.

VI. REFERENCES

[1] "Final Group Project Machine Learning," Dalhousie University. [Online], Available: https://dal.brightspace.com/d2l/home/248907 [Accessed: March 28, 2022].

[2] "Random Forest Regression," Medium. [Online], Available: https://towardsdatascience.com/random-forest-regression-5f605132d19d [Accessed: March 28, 2022].

[3] "Random Forest Regressor," scikit learn.[Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html [Accessed: March 28, 2022].

[4] "Sklearn Neural Network Example – MLPRegressor ," Data Analytics. [Online], Available: https://vitalflux.com/sklearn-neural-network-regression-example-mlpregressor/ [Accessed: March 28, 2022].

[5] "MLPRegressor," sckit learn.[Online], Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html [Accessed: March 28, 2022].

[6] "Repeated k-Fold Cross-Validation for Model Evaluation in Python," Machine Learning Mastery. [Online], Available: https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/#:~:text=Different%20splits%20of%20the%20data,all%20folds%20from%20all%20runs. [Accessed: March 28, 2022].

[7] "The Complete Guide to Time Series Data," Clarify. [Online], Available: https://www.clarify.io/learn/time-series-data [Accessed: March 28, 2022].

[8] "Cross-Validation in Machine Learning: How to Do It Right," Neptune.ai. [Online], Available: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right [Accessed: March 28, 2022].