T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

REMAINING USEFUL LIFETIME PREDICTION FOR PREDICTIVE MAINTENANCE IN MANUFACTURING

Master Thesis

BERNAR TAŞCI

İSTANBUL, 2021

T.C.
BAHÇEŞEHİR UNIVERSITY

GRADUATE SCHOOL,
COMPUTER ENGINEERING MASTER'S PROGRAM

REMAINING USEFUL LIFETIME PREDICTION FOR PREDICTIVE MAINTENANCE IN MANUFACTURING

Master Thesis

BERNAR TAŞCI

Thesis Advisor: ASSIST. PROF. SERKAN AYVAZ

İSTANBUL, 2021

T.C.
BAHCESEHIR UNIVERSITY
GRADUATE SCHOOL

...../...../.....

MASTER THESIS APPROVAL FORM

Program Name:	Computer Engineering Master's Program
Student's Name and Surname:	Bernar TAŞCI
Name of the Thesis:	Remaining Useful Lifetime Prediction for Predictive Maintenance in Manufacturing
Thesis Defense Date:	23.06.2021

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Assoc. Prof. Dr. Burak KÜNTAY
Institute Director

This thesis was read by us, quality and content as a Master's thesis has been seen and accepted as sufficient.

	Title/Name	**Signature**
Thesis Advisor's	Assist. Prof. Serkan AYVAZ	
Member's	Assist. Prof. Tamer UÇAR	
Member's	Prof. Selim ZAİM	

ACKNOWLEDGMENT

First of all, I owe a great thanks to my supervisor Dr. Serkan Ayvaz for his outstanding support and guidance through my master's program and thesis study. He has been a mentor for me, and I am very grateful for having a chance to work with him.

I would like to thank Mr. Ammar Omar for his dedication and support throughout the whole study.

I would also like to thank my family for their encouragements and support.

Last but not least, I would like to thank my wife, Hena, for all of her unlimited and unconditional support and belief in me.

Istanbul, 2021 Bernar Taşcı

ABSTRACT

REMAINING USEFUL LIFETIME PREDICTION FOR PREDICTIVE MAINTENANCE IN MANUFACTURING

Bernar Taşcı

Computer Engineering Master's Program

Thesis Supervisor: Assist. Prof. Serkan Ayvaz

June 2021, 44 Pages

In this study, a predictive maintenance approach is proposed by predicting RUL of a real-life production line. A detailed review of the literature is conducted, and various aspects of different methodologies have been explained. By using machine learning algorithms, possible future failures are tried to detect, and an early maintenance task is scheduled for preventive maintenance. Although the first trials showed promising results, with the implementation of newly generated data, some challenges were addressed. In contrast to unsuccessful predictions, the detailed analysis indicated noteworthy results, which shifted the focal point of the study where new evaluation methods are tried.

Keywords: Predictive Maintenance, Machine Learning, Remaining Useful Lifetime, Manufacturing.

ÖZET

ÜRETİM HATLARINDA KESTİRİMCİ BAKIM İÇİN KALAN FAYDALI ÖMÜR TAHMİNİ

Bernar Taşcı

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Serkan AYVAZ

Haziran 2021, 44 Sayfa

Bu çalışmada, gerçek bir üretim hattı için kalan faydalı ömür tahmini ile kestirimci bakım yaklaşımı önerilmiştir Konu ilgili detaylı literatür taraması yapılmış ve farklı metotlara ait görüşler açıklanmıştır. Olası arızalar makine öğrenimi algoritmaları vasıtası ile yakalanmaya çalışılmıştır ve bu sayede istenmeyen hatalardan kaynaklanan üretim kesintileri önlenmeye çalışılmıştır. İlk denemelerde elde edilen başarılara karşın, sonradan sağlanan veri seti üzerinde yapılan çalışmalar sonucu beklenmedik sonuçlar elde edilmiştir. Bu sonuçların yanı sıra, detaylı araştırmalar umut vaat eden sonuçları açığa çıkarmıştır ve bu da çalışmadaki odak noktasını faklı değerlendirme yaklaşımları üzerine yoğunlaşmaya doğru yönlendirmiştir.

Anahtar Kelimeler: Kestirimci Bakım, Makine Öğrenimi, Kalan Faydalı Ömür, Üretim.

CONTENTS

TABLES

FIGURES

INDEX OF ABBREVIATIONS

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
ARIMA	:	Autoregressive Integrated Moving Average
BDT	:	Boosted Decision Tree
CNN	:	Convolutional Neural Network
CRBM	:	Conditional Restricted Boltzmann Machine
DCNN	:	Deep Convolutional Neural Network
DJ	:	Decision Jungle
DNN	:	Deep Neural Network
FCRBM	:	Factored Conditional Restricted Boltzmann Machine
GRU	:	Gated Recurrent Unit
IIoT	:	Industrial Internet of Things
IoT	:	Internet of Things
LOF	:	Local-outlier-factor
LSTM	:	Long Short-Term Memory
MAE	:	Mean Absolute Error
MAPE	:	Mean Absolute Percentage Error
MLP	:	Multilayer Perceptron
NN	:	Neural Network
PCA	:	Principal Component Analysis
PdM	:	Predictive Maintenance
R^2	:	R-squared
RF	:	Random Forest
RMSE	:	Root Mean Squared Error
RNN	:	Recurrent Neural Network
RUL	:	Remaining Useful Lifetime
SVM	:	Support Vector Machine
SVR	:	Support Vector Regression
WCSS	:	Within-Cluster Sum of Square
XGB	:	XGBoost (eXtreme Gradient Boosting)

1. INTRODUCTION

Every day manufacturing extends its place in our daily life. With the rapid increase in population resulted in the need for complex manufacturing lines to meet the demand. As the manufacturing began to shift with the industrial developments, more intelligent and more complex machines found their place in production lines, creating smarter ecosystems. But maintenance of such complex structures became vital for reliability and sustainability.

Current innovations in technology started shifting the traditional behavior of human-machine interaction. With the increased processing power of machines and algorithms, this interaction turned from one way to mutual with the help of Artificial Intelligence (AI). Such shift not only found its place in the daily life but also in the industry and production areas in terms of automation. In order to remain competitive, firms started to rely on automation more and more to increase productivity (Wright & Schultz 2018).

With the advancements in machines and the ability of device-to-device communication, Internet of Things (IoT) emerged to production lines providing active monitoring of the system. This shift in the industry shaped a new phase, namely Industry 4.0 (J. Lee et al. 2018). Such advancement reveals what is called big data in the industrial facilities that need to be handled in a systematic manner. Such data shows the ability of system coordination and feedback among devices which provides the capability of self-calibration and system performance improvements (Wang et al. 2016).

While conventional maintenance approaches continued until the recent years, integration of AI and Industrial Internet of Things (IIoT) in the manufacturing formed a new phase where Predictive Maintenance (PdM) become the main objective. With the implementation of PdM, scheduled maintenance approaches led the way to efficient use of machinery until their useful lifetime. This removed the effects of unintentional and unnecessary stops for maintenance purposes, creating a cost-saving, high-efficiency, fully optimized manufacturing perspective.

This project aims to create a PdM system to capture future failures and send warnings before their occurrence with the use of IoT sensor readings and machine learning algorithms. Predicted warnings will prevent unnecessary waste caused by the time lost and restart duration, as the line will not be able to produce usable goods as soon as it starts functioning again. To create such a system, different methods and approaches have been implemented on the dataset and compared to obtain the most accurate option.

The other parts of the study are listed as follows: in the second chapter, literature is reviewed for similar approaches conducted before and explained briefly. In the third chapter, the dataset used for the study is introduced, and the necessary processing steps are described in detail. In the fourth chapter, methodologies that are used for creating such a system are explained, and in the fifth chapter, results for proposed methods are evaluated. In the sixth chapter findings of this study are described. With the final chapter, the research is summarized, and the plans for future work are listed.

2. LITERATURE REVIEW

2.1 STATISTICAL APPROACHES TO PREDICTIVE MAINTENANCE

PdM plays a significant role in the area of industry and production due to the costs caused by the unplanned, fault-related stops in either facilities or working machines. In their study, Shimada & Sakajo proposed a statistical approach for maintenance scheduling is proposed. Their approach mainly focused on the timing of the maintenance after the warning delivered by the model. This was developed by using a characteristic index extracted by the cumulative failure probability distribution. An approach from warning state to failure state analysis could reduce the burden of maintenance workers as well as increase the performance of model predictions by estimating the time required for maintenance after model warning occurrence. Their approach managed to reduce the failure ratio by 12 percent (Shimada & Sakajo 2016).

Ho et al. focused on Autoregressive Integrated Moving Average (ARIMA) and Recurrent Neural Network (RNN) models to predict failures. To model a nonseasonal time series, combination of past values and errors are used in ARIMA. Their findings prove that both methods perform better at short-term dependencies. However, one comparison shows that feed-forward neural networks performed poorly compared to ARIMA and RNN (Ho et al. 2002).

In another study, Kanawaday & Sane proposed ARIMA forecasting to predict failures and quality defects on a data collected from real-world Slitting Machines. The first phase of their two-stage approach starts with data analysis, clustering and supervised learning methods in order to obtain information the data followed by predictive model implementation with the use of ARIMA. Their architecture consists of two stacked systems, one being ARIMA to predict the parameter values of the upcoming cycles and the other being the supervised models that predict the outcome if a cycle is bad or not. In this study, Deep Neural Network (DNN) proved to perform better compared to the rest (Kanawaday & Sane 2017).

Another study showed the use of ARIMA for trend analysis and prediction process. The proposed methodology uses ARIMA for trend analysis in the time series data. Later the predicted values are passed through Principal Component Analysis (PCA) for feature reduction by eliminating highly correlated variables. Finally for maintenance prediction approach Support Vector Regression (SVR) algorithm is used (Francis & Mohan 2019).

Although the performance of statistical approaches are pointed in the literature, it is known that when it comes to long-term dependencies, Long Short-Term Memory (LSTM) tends to perform better compared to ARIMA. It is also observed that LSTM and Gated Recurrent Unit (GRU) significantly outperforms ARIMA while producing similar predictions compared to ARIMA (Karpathy et al. 2015).

2.2 DATA-DRIVEN AI APPLICATIONS TO FAILURE PREDICTION

In their study, Zhang et al. (2018) used NASA's C-MAAPSS engine dataset for system performance degradation prediction by revealing the temporal dependency of the sensor data. They exposed the effect of temporal-dependency of the data, on predicting the engine degradation (Zhang et al. 2018b). For model training and prediction, 6 of the sensors are chosen, those being the most commonly equipped among other aircraft engines. Due to the nature of time-series data, deep learning-based models are preferred for capturing data dependency. The main focus of the study is the use LSTM based models for prediction. This is due to the fact LSTM is more capable of remembering long-term dependencies, it is better at capturing patterns in the dataset. Evaluation of the proposed model is concluded using synthetically generated 200 curves. The performance of the proposed LSTM model is compared to various ML algorithms like SVR, Multilayer Perceptron (MLP), Relevance Vector Regression (RVR), and Deep Convolutional Neural Networks (DCNN), where LSTM model significantly outperformed the rest. It is also shown in the study that if the prediction curve decreases, model prediction performance increases respectively (Zhang et al. 2018a).

W. J. Lee et al. studied PdM algorithms in two different datasets, one being the cutting tool and the other spindle motor wear dataset. The milling tool wear prediction is vital for

cutting performance, and the Remaining Useful Lifetime (RUL) prediction is required to prevent improper production as well as the waste of material. Spindle motor being a rotating element, similar to milling tool, it is an essential component of production quality. Both datasets are collected until the occurrence of a system failure. Two different classification algorithms, SVM and Convolutional Neural Network (CNN), are applied for wear prediction, and for the performance evaluation, a confusion matrix is drawn where the predictions are classified into three groups as follows: Normal, warning, and failure. They concluded that frequency-domain results using CNN provided high accuracy prediction results (W. J. Lee et al. 2019).

Similarly, Traini et al. provided a framework solution in their study for milling cutting-tool maintenance. Their approach combines both regression and classification algorithms to create a framework on analog sensor readings to improve the human-machine relationship in terms of PdM. Regression algorithms used in this study are; Linear Regression (LR), Decision Forest Regression (DF), Bayesian Linear Regression (BLR), Boosted Decision Tree Regression (BDT), and Neural Network Regression (NN). Classification algorithms are; Logistic Regression (LogR), DF, Decision Jungle (DJ), BDT, Neural Network (NN). The regression algorithm is applied on both flank wear (*VB*) and RUL values where the classification is applied under two classes regarding Safe and Worn. Best results for regression are obtained on NN regression models with the highest R^2 value and best results for classification are obtained using Two-Class BDT with the highest accuracy using confusion matrix for evaluation (Traini et al. 2019).

In their recent study, Rivera et al. focused on the prediction of a vital component inside an injection molding machine. In contrast with their previous study (Rivera et al. 2018), current work focuses on the same data with a top-down approach where the expert knowledge is omitted from the preparation. Different than previously reviewed studies, their prediction approach relies on anomaly detection based on kernel density estimation, which is a statistical method. Without the knowledge of underlying machine physics by using the whole raw data, after applying PCA, they fed the data into a local-outlier-factor (LOF) algorithm to detect unusual cycles. For this problem, their choice of threshold for LOF algorithm is set to be three. In their statement, they pointed out that the reason behind

the anomaly declaration of the algorithm is not clear, and deciding whether to omit it from the future analysis is true or not. But the detection of hidden patterns among variables is stated as one of the pros of this approach. They also pointed out the need for almost infinite computational power via cloud computing would turn out to be very useful in such a scenario when the cyber-security concerns were refused to take notice. In conclusion, the importance of expert knowledge is pointed out as the reduction of unnecessary data could increase performance by decreasing computational efforts (Rivera et al. 2018).

Li et al. introduce a data mining-based fault diagnosis framework in their study. The framework consists of five modules. Their research is collected from a remote customer site where online machine condition monitoring is applied with the use of fault prediction and maintenance strategy optimization. As a result, maintenance implementation and error correction is put to practical use. The study focuses on the prognosis for a backlash error. In the data mining phase, they introduced an Artificial Neural Network (ANN) to predict the upcoming errors using the data of the past three weeks. The proposed ANN model has the advantage of fault tolerance, adaptability, and generalization. But on the other hand, the model is prone to the effect of various factors. Similar to Rivera at al., they pointed out the importance of expert knowledge for the development of an accuracy model (Li et al. 2017).

Finally, in their most recent study, Ayvaz & Alpay proposed a PdM framework on a real-world assembly line that produces customer hygiene products. Their study covers the stages of data collection and processing, model development and implementation, and error reporting for PdM purposes. They emphasize the scalability of the proposed architecture by pointing out the performance of applied algorithms. Moreover, another critical achievement is mentioned in terms of the digital transformation of the facility. Ensemble learning algorithms used, namely Random Forest (RF) and XGBoost (XGB), obtaining R^2 score of 0,982 and 0,979 respectively proves to be notable (Ayvaz & Alpay 2021).

2.3 DEEP LEARNING APPLICATIONS IN PREDICTIVE MAINTENANCE

Xie et al. pointed out the importance of deep learning in applications on IoT and Smart City structures where time-series data is present. They stated the prediction performance of LSTM networks over such data due to their ability to maintain long-term dependencies. In their use case, they developed and used an LSTM structure combined with a Gaussian Naïve Bayes model to detect anomalies on predictions. Using the three different real-world datasets, they were able to display the performance of the proposed model compared with the LSTM NN model and MLP model (Xie et al. 2021).

In another study, Mocanu et al. provided a research on building energy consumption prediction based on two stochastic deep learning models, namely Conditional Restricted Boltzmann Machine (CRBM) and Factored Conditional Restricted Boltzmann Machine (FCRBM). In their work, they compared the results of five different models, and results proved that FCRBM outperformed Support Vector Machine (SVM), RNN, ANN, and CRBM. One highlight of their study shows that all methods performed better in predicting the aggregated active power consumption rather than sub-meterings (Mocanu et al. 2016).

In the field of renewable energy power forecasting, Gensler et al. provided a study on powerful deep learning algorithms compared to other reference models. Their findings showed that Deep Belief Networks (DBN) and a combination of AutoEncoder and LSTM, namely Auto-LSTM, performed well on solar power forecast (Gensler et al. 2016).

Rieger et al. provided a review in the field of IIoT where deep learning algorithms are used to provide real-time processing for PdM purposes (Rieger et al. 2019).

From a different perspective, Chen et al. focused on challenges using PdM on censored data, which are partially labeled. Their study investigates a new method called Cox proportional hazard deep learning (CoxPHDL), which establishes a time-between failure (TBF) prediction model based on previously collected data. This approach contains three stages, where the first phase converts nominal data into binary data with a one-hot encoding approach followed by an autoencoder for further processing the binary data.

The second phase feeds censored data from the newly generated data, which is the combination of the output of the autoencoder and numeric data, into the Cox proportional hazard model (CoxPHM) for data generation. Finally, the processed data is used to train the LSTM network. Compared to RNN, ANN, DCNN, and SVM, LSTM obtained the best results (Chen et al. 2020).

Mohammadi et al. conducted a large-scale survey on DL for IoT big data in the literature, where they first distinguished the difference between IoT big data analytics and IoT streaming data analytics. They point out the difference between IoT data and big data, where IoT possesses the characteristics of large-scale streaming, heterogeneity, time-space correlation, and a high amount of noise. Their findings show the frequency of models used in different studies where CNNs have been used around 43 percent of the papers followed by LSTMs and RNNs with a 30 percent ratio (Mohammadi et al. 2018).

Finally, Canziani et al. covered the performance requirements in a real-world DNN application. Their findings provide a detailed comparison of the performance requirements of a NN application while preserving efficiency. In their study, they observed the relationship between accuracy and computational cost where a small amount of increase in the accuracy resulted in progressive increment on computational time (Canziani et al. 2017). These findings line up with the approach in our study where computational power requirements and processing times are taken into consideration in terms of model selection. Due to their feasible use in real-time prediction applications, data-driven machine learning algorithms are evaluated in this study.

2.4 PREDICTIVE MAINTENANCE IN SUSTAINABILITY OF CYBER-MANUFACTURING

From the manufacturers point of view, with the increasing pressure of sustainability and demand on eliminating environmental effects of production, it is getting vital to achieving such goals while protecting competitiveness (Song & Moon 2018). The constructive impact of PdM is not only marked by the manufacturing perspective but also drawn attention from the topic of sustainability. It is pointed out that with the help of PdM, the

availability and efficiency of the resources can be increased; as a result, general costs can be decreased. Using the resources efficiently will enhance environmental sustainability as well (Jasiulewicz-Kaczmarek et al. 2020). A major part of the sustainability is linked to industry 4.0 implementations, where usage of big data in PdM approaches plays a big role (Machado et al. 2020).

3. DATA EXPLORATION AND PREPARATION

In this chapter, the dataset provided for this study is introduced, and the stages necessary for data preparation are explained.

3.1 GENERAL INFORMATION OF DATA

There are two necessary datasets used in this study which are sensors data and stops data. The dataset was collected from a real-world assembly line producing consumer hygiene goods. Inside the production line, several IoT sensors are being used, which send out sensor readings. Those sensor readings change over time based on some properties like weight, speed, temperature, electrical current, vacuum, and air pressure. Each sensor sends data in periods of 3 to 6 seconds.

3.2 DATA COLLECTION

The sensor dataset was collected and stored in a database for one year. This dataset contains 101 features and 8,668,431 rows. There are 50 sensors and 50 timestamps related to each sensor in this dataset. There is also a final feature called Var1, which represents the status of the line where the value zero represents that the line is working and any other non-zero value denotes that the line is not working. This variable is used as a control mechanism in the upcoming works.

The real-time production line does not provide RUL values. The features stored in the dataset are separated into two main groups, sensor readings and the timestamp of the reading for each sensor. As a result, this prediction problem cannot be addressed by a time series approach since no RUL value is present in the dataset.

The second file used in this study is the stops dataset. Stops dataset consists of 34 columns and 6787 rows. This file is created at the production line, where any instances of stops are noted for one year with various details like the cause, time, duration, team, line, status

(if planned or not), and comments. The information necessary for RUL creation process is as follows:

i. *StartTime*: Start time and date of stop instance.
ii. *Downtime*: Duration of stop in seconds.
iii. *Cause*: Cause of the stop.

There are 27 unique causes of stops in the stops dataset, and 228 rows of those are related to a single error which is the most occurring error in the production line to be prevented.

For this particular study, the two datasets provide a projection on the distribution of stops for a 1-year dataset. Only 228 stops for a year is a small amount of number for model training, and this makes it very hard to predict tasks and creates challenges.

3.3 CREATION OF RUL VALUES

A straightforward approach for this study could be a classification problem but predicting the failure as a classification problem is not convenient since the purpose is to prevent the occurrence of the failure, not receiving an alert at the stop instance. Therefore, the situation is handled as a regression task which will allow for necessary actions to be taken before the failure occurs and preventing unnecessary stops in the production line. As a result, the preprocessing approach starts with the creation of necessary RUL values for the dataset. Considering the size of the data, SQL is capable of handling such a process. Using SQL, necessary features are selected and transferred to a new table, where the RUL values are created using the stops data.

The process of RUL creation starts by creating the MaxDate variable for each row in the sensors data. There are 50 timestamps on each row representing the time of sensor readings, and when checked, it is observed that these timestamps on each row are very close to each other. So the maximum value among timestamps is selected as the MaxDate. Next step is creating UnixDate, BeginTime, and EndTime from the stops data. UnixDate is created by converting the StartTime feature, which contains time and date information,

into UNIX timestamp. BeginTime is created by subtracting 3 minutes (180 in seconds) from UnixDate because the actual time of the stop is usually 3 minutes before the noted time. EndTime is created by adding the Downtime into BeginTime where downtime is stored in seconds, so it is converted to UNIX date as well.

The next step is ranking the instances to find the exact instance of stop by calculating the difference between each stop's UNIX date and the MaxDate variable created from the sensors data. After this step, each instance of stop is marked. The process continues with creating and assigning values to Error, ExactError, NextErrorDate, and RUL variables. Error and ExactError variables are assigned zero, where NextErrorDate and RUL variables are assigned null for each instance. The records between StartTime and Endtime of stops are updated where Error variable is assigned 1 and the RUL value for corresponding instances are marked zero since during the stop RUL cannot have any value other than zero. Next the records nearest to StartTime are updated where ExactError variable is assigned 1.

To finalize the cleanup, a final variable NextErrorDate is introduced and updated. This variable is marked using the closest error date to that instance of stop where Error variable is zero. In the end the subtraction between this variable and MaxDate provides the RUL values for every instance where RUL no value has been assigned. By selecting the 18 sensor readings which are related to the error, a new dataset is created as shown in the Table 3.1. After RUL creation, the final dataset is explored for future analysis.

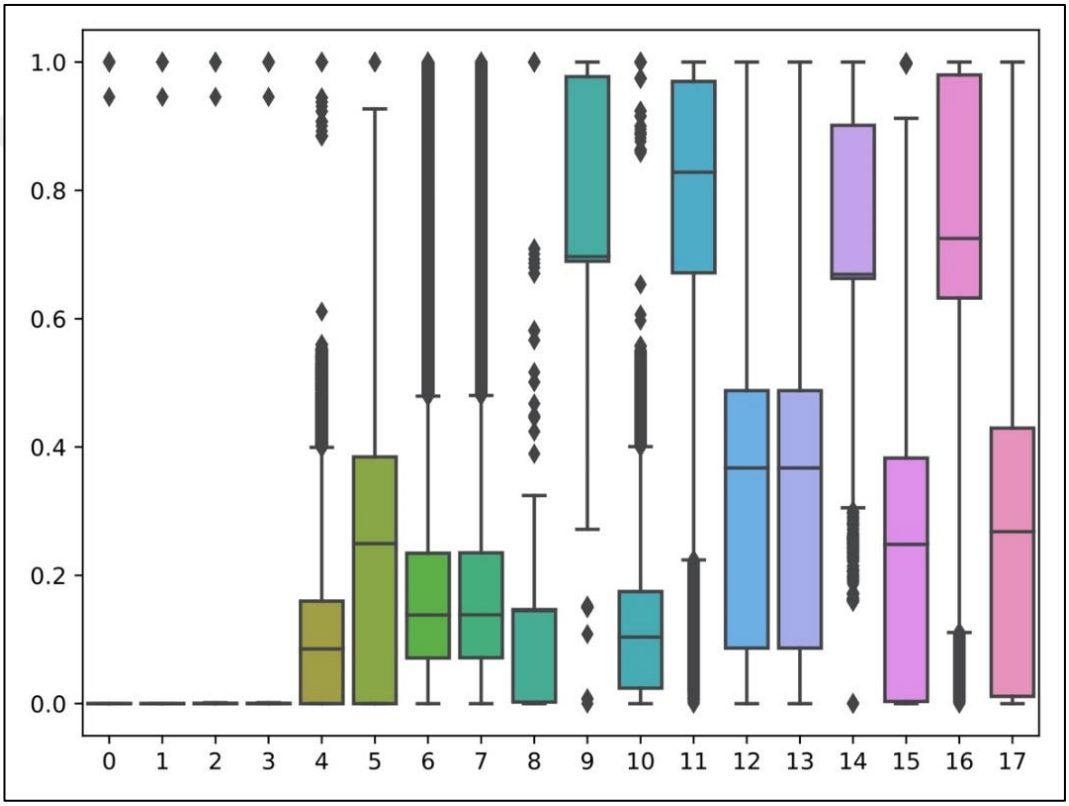
Table 3.1: Output of newly generated dataset

Var1	Timestamp	18 Sensors Readings (a1-a18)	RUL

3.4 NORMALIZATION

The data distribution of sensor values are shown in the Figure 3.1. Here we can see the distribution of the sensor values after min-max normalization. As the nature of IoT sensor data, the dataset is very unbalanced. During the model fitting, the dataset is used after scaling, where StandardScaler algorithm of scikit-learn library is used.

Figure 3.1: Boxplot of dataset feature distribution



3.5 SPLIT TRAIN AND TEST SET

To avoid overfitting and underfitting, the most important step before modeling is splitting the dataset into training and test sets. Using the training set model fitting is performed and to avoid memorization of the dataset and to obtain reliable results, model is tested using the test set.

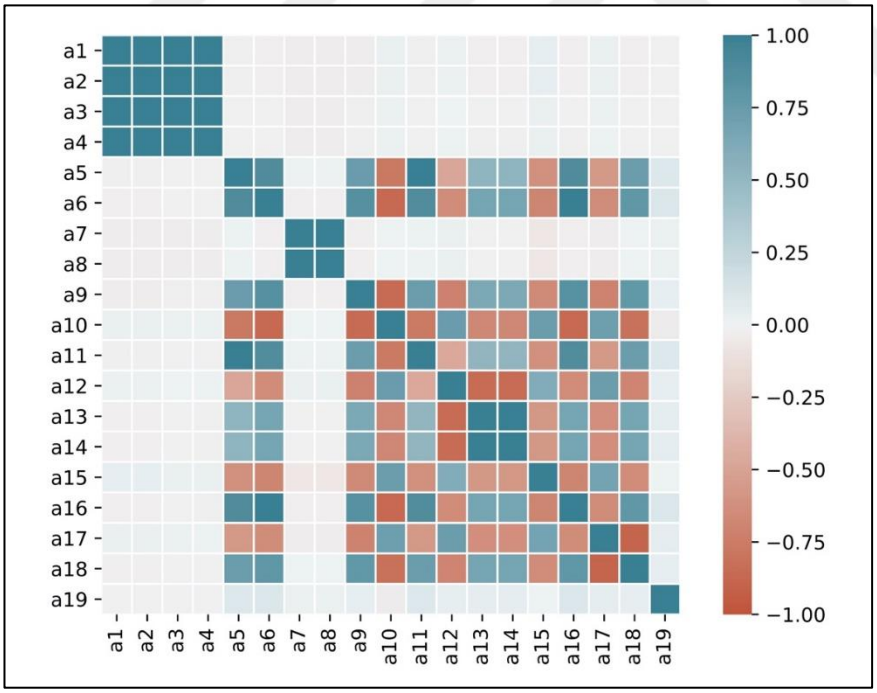
Durin hyperparameter tuning of the model training, 5-fold cross validation is applied to the dataset to avoid overfitting.

To obtain comparable results for each conducted test, same seed is used to split the data into train and test sets. In our study 70 percent of the dataset is used to train the model and 30 percent of the dataset is used to test the model performance.

3.6 CORRELATION

Although the selected sensors for the prediction approach are known for their direct connection with the related error, for detailed exploration purpose correlation analysis has been applied. Figure 3.2 shows the correlation plot, where highly correlated features are observed.

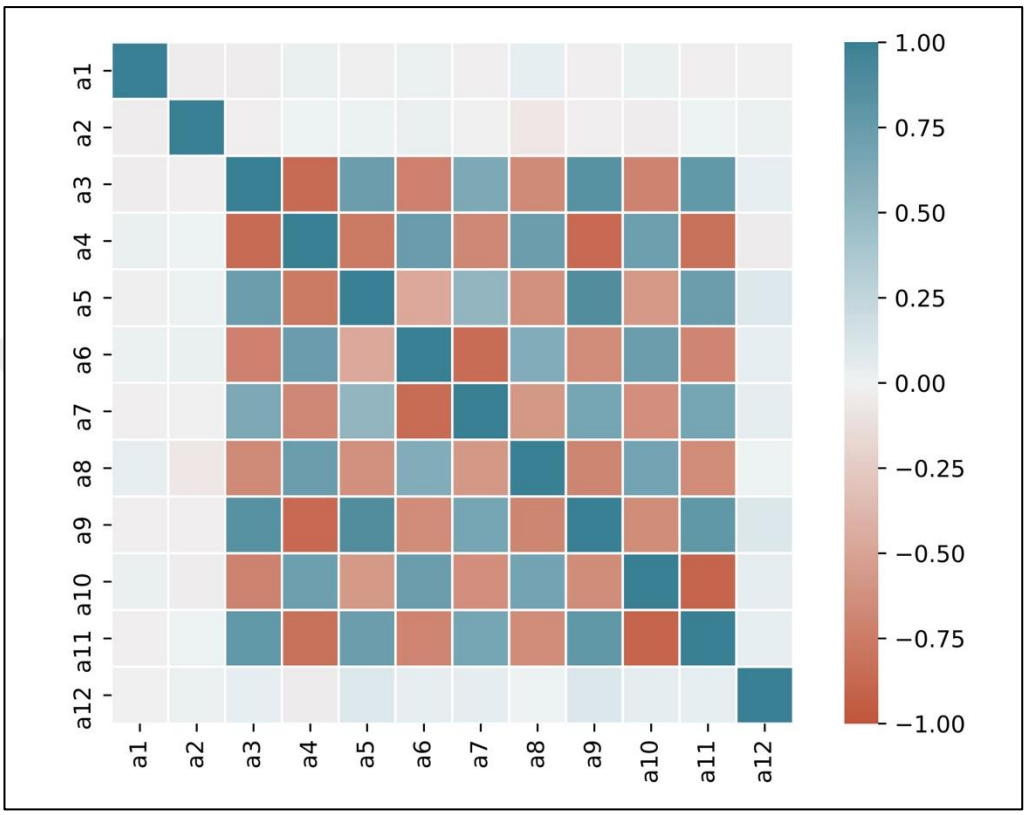
Figure 3.2: Correlation plot of the dataset



In our case this is expectable since some of the sensors carry out information from same locations where multiple sensors are placed with different viewing and measuring angles for detailed monitoring. For comparison purposes another dataset has been created where

highly correlated features are removed from the dataset as shown in the Figure 3.3. This new created dataset contains eleven features while the original dataset has eighteen.

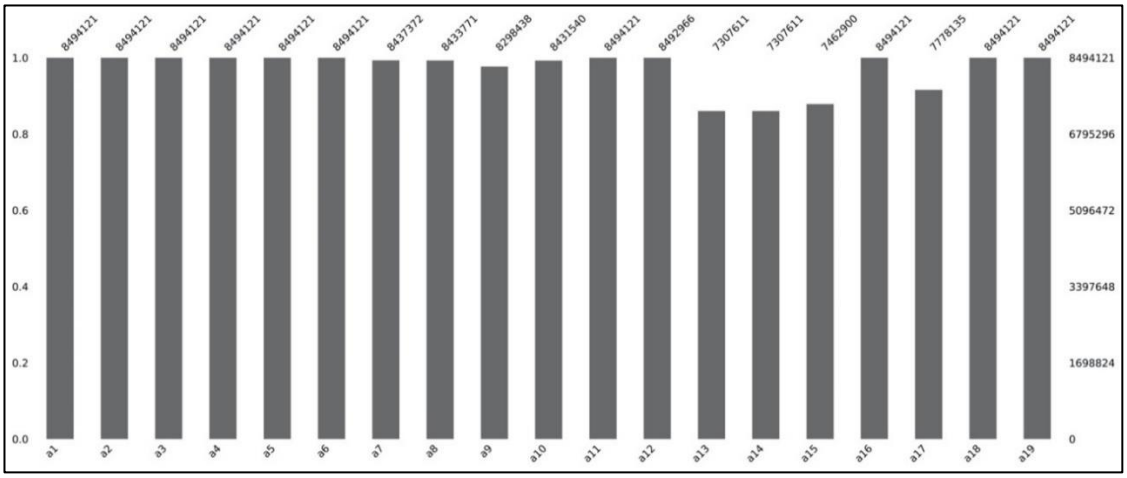
Figure 3.3: Correlation plot after removing highly correlated features



3.7 MISSING VALUE

As observed from the Figure 3.4, vast majority of the data is filled where only one percent of the data is missing. Missing values are filled with median values of the related column before model fitting.

Figure 3.4: Missing value plot of the dataset



3.8 TIMESTAMP CONVERSION

The database which contains sensor readings, provides UNIX timestamps where the stops dataset contains time information in local time domain (Turkey GTM+3). This detail is addressed after the first trials of the tests where poor results are observed and investigated in a detailed manner. The actual stops from the stops dataset and the newly created dataset were not matching up. We were able to overcome this problem by converting the local time of stops into UNIX timestamps. This detail is addressed with the use of Var1 feature explained in the previous headlines.

3.9 ALTERNATIVE DATASET – STOPS REMOVED

After each stop occurs, until the end of the stop duration the dataset has zero value for RUL which is produced after the SQL cleanup script. But during the stops, sensors in the production line keep sending readings based on the current status and this could be a misleading input for the model trained. For this matter, every row that has zero value after the initial stop is removed from the dataset before training the model.

4. METHODS & ANALYSIS

Models in this study are created using Python programming language. The machine learning algorithms chosen for model creation are from scikit-learn and xgboost libraries.

4.1 CONSTITUENT MODELS

4.1.1 Support Vector Regression

Being a supervised learning approach, it is a popular tool for non-linear problems as it is in this study, higher dimensional spaces can be used to map the data to obtain higher performance. In our case Radial Basis Function (RBF) kernel is used based on comparison with the linear SVR.

4.1.2 Multilayer Perceptron

MLP is type of a feed forward NN that contain three layers, namely input layer, hidden layer and output layer. It is strong at solving problems which are not linearly separable. Results of hyperparameter tuning showed that using relu activation function, adam optimizer, 50 hidden layer nodes and epsilon of 1e-08 fit the dataset well.

4.2 ENSEMBLE LEARNING

Ensemble learning algorithms are the combination of multiple weak learners. While these individual algorithms lack performance, the output of multiple weak learners creates a well-performing prediction model. These algorithms are classified into three groups, Bagging, Boosting and Stacking. The following subsections describe the ensemble learning algorithms that are used in this study.

4.2.1 Random Forest

RF is a bagging ensemble method, which is kind of a decision tree algorithm that can be applied either in classification or prediction problems. The main problem with the decision trees is the tendency of overfitting the data. RF handles this problem by creating subsets of data and training each with different tree nodes. Each tree in RF makes a prediction and the resulting prediction is the average of all the predictions created by the decision trees.

RF can either be applied as a classification or regression model. In this study, RF model was created using Python language RandomForestRegressor background in scikit-learn machine learning library. Hyperparameter tuning was applied by random search with 5-fold cross validation and the number of estimators was selected 51 as the best performing parameter for model training.

4.2.2 eXtreme Gradient Boosting

XGB is a type of optimized Gradient Boosting algorithm. Compared to RF, XGB creates trees with fewer splits and so it can overcome the overfitting problem. Due to its precision and efficiency, it is widely used in prediction problems. Results of hyperparameter tuning showed that XGB performed best at maximum depth of 5, learning rate of 0,3 and number of estimators was selected 100.

4.3 CLUSTERING

Due to the nature of the problem, it is essential to predict the RUL value rather than the class value of the instance as an error or not. But, in order to add some meaningful information to the dataset, clustering is applied. The output of clustering could provide additional information for the prediction approach in predicting the RUL value.

For determination of number of clusters, Elbow method is applied. This method simply creates and calculates each number of possible clusters (k) from one to ten and for each

k Within-Cluster Sum of Square (WCSS) is calculated. WCSS represents the sum of distances between centroid of the cluster and each point inside. The resulting plot provides a graphic similar to an elbow, where the method is taken its name, and the point where graph starts to continue parallel to x axis is representing the optimal number of clusters.

To perform clustering, KMeans algorithm from scikit-learn library is used.

4.4 EVALUATION

For the evaluation of the model performance results, four measuring techniques are applied. First metric is coefficient of determination, also known as R-squared (R^2), is a widely used parameter for performance evaluation in statistics. It represents the proportion of variance which is explained for a dependent variable (Equation 4.1).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \quad \textbf{(4.1)}$$

Mean Absolute Error (MAE) measures the average difference between actual and predicted values by eliminating their direction (Equation 4.2).

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \quad \textbf{(4.2)}$$

Mean Absolute Percentage Error (MAPE) represents the accuracy as a percentage. Main advantage of this metric is the use of absolute error, since it avoids cancelation of different signed errors (Equation 4.3).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad \textbf{(4.3)}$$

Root Mean Squared Error (RMSE) is a metric used to describe how far the predictions are distributed from the actual values. In other words it represents the concentration of the predictions against the best fit (Equation 4.4).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad \textbf{(4.4)}$$

Finally, to help better explain the model performance results, inverse MAPE is calculated as shown in Equation 4.5 to help explain the accuracy of the predictions.

$$Inverse\ MAPE = 100 - MAPE \quad \textbf{(4.5)}$$

4.4.1 Median Filter

Median filter is type of a non-linear filter, known for its performance in reducing the noises in the dataset. In this study it is used to eliminate the random spikes from the prediction results and smooth out the output.

Medfilt algorithm from SciPy library is used to perform median filter on the prediction's dataset. In this approach kernel values describe the size of windows to calculate Kernel values used for filtering are 10, 50 and 100.

4.4.2 True Positive – False Positive Approach

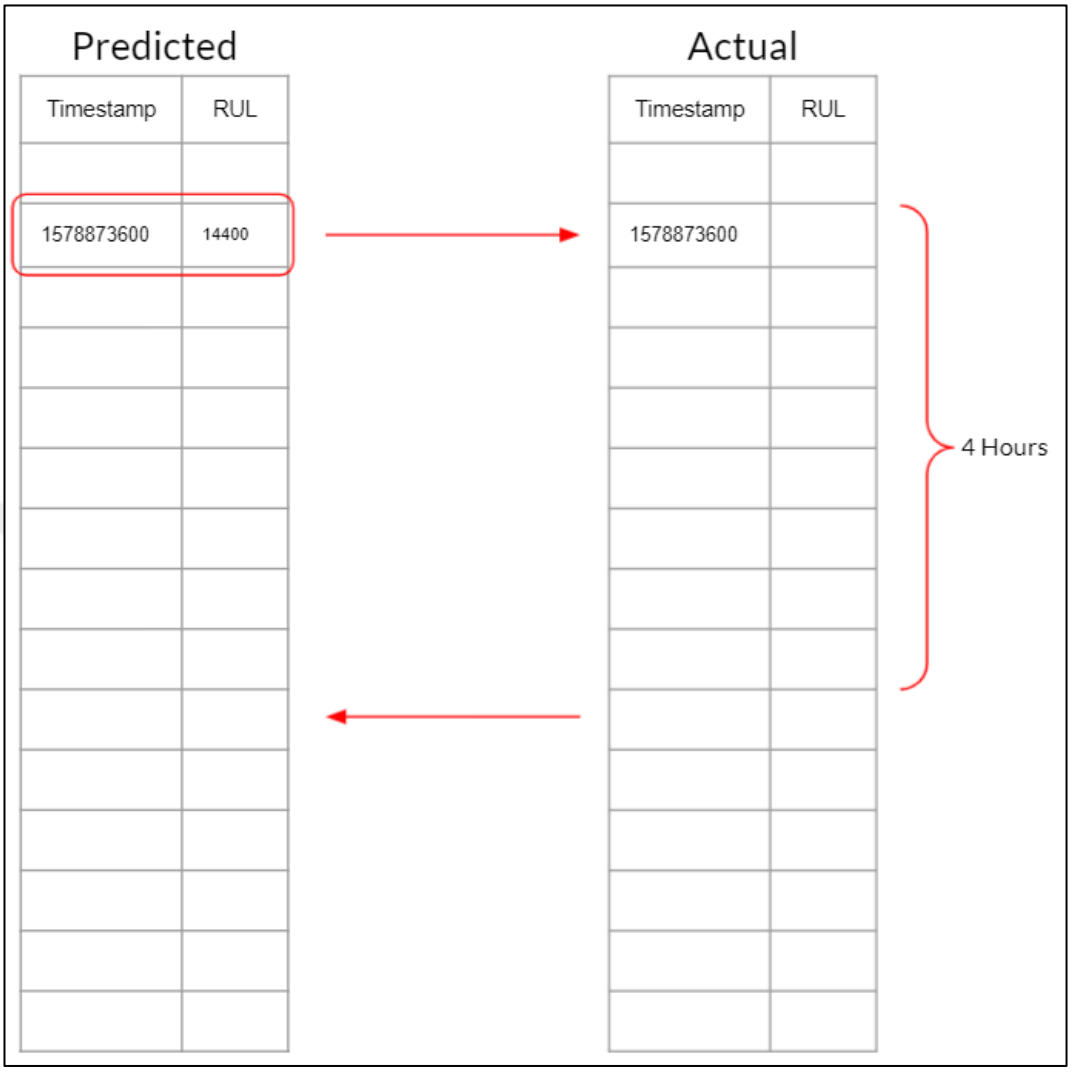
In this new approach, we define "True Positive" (TP) and "False Positive" (FP) as our new assessment method. As shown in Figure 4.1, if the prediction model predicts RUL value of 14400 (4 hours in UNIX time), the timestamp of that instance is flagged and using the actual dataset starting from that exact timestamp until 4 hours is passed, it is checked if a stop occurred or not. If so, it is marked as "True Positive" otherwise it is marked "False Positive". In order to check the results, actual label values (RUL values) are controlled.

The main aim of this approach is focusing on decreasing the False Positive values as well as increasing True Positive values. Due to the characteristic of the dataset, providing a regression solution is not sufficient. When the framework point of application is designed, there is no global true or false. Problem specific solutions take a major role in the industrial RUL prediction.

For the reliability of the predictions, it is essential for the production line to decrease false alarms which causes unnecessary actions, as a result, stops for system maintenance purposes.

During the production, the system is set to send alarms if it detects 4 hours or fewer RUL values. Considering the results of the prediction, a new evaluation for the performance is tried where the goal is to decrease the FP alarms as well as change the way alarms are sent during production.

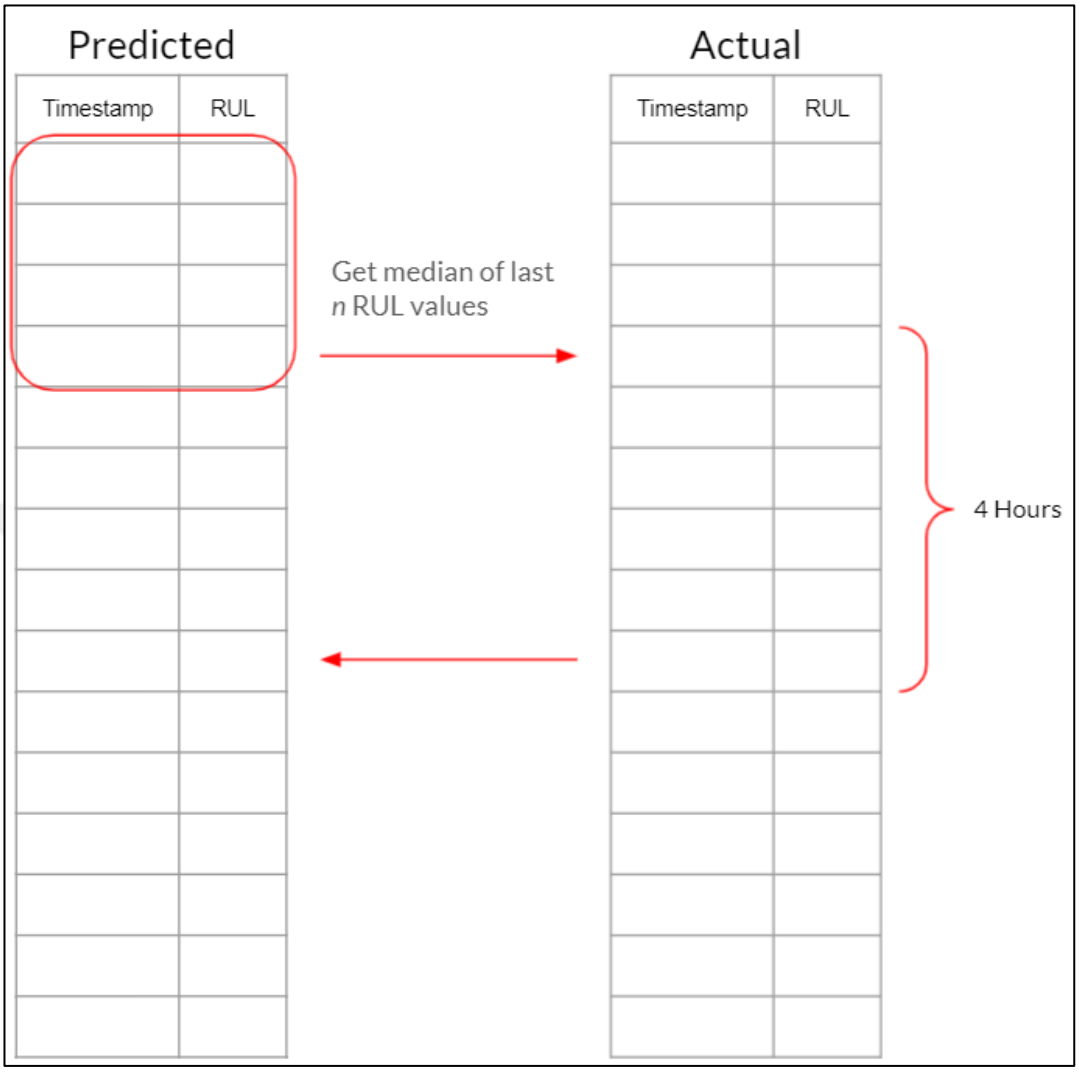
Figure 4.1: TP-FP approach script diagram



4.4.3 True Positive – False Positive Approach with Median Filtering

In order to decrease the effect of the high variation of predicted RUL values, we changed the way of choosing the RUL value. Rather than choosing the predicted RUL value from the prediction, we started taking the median of the last n number of columns where n is tested for different values.

Figure 4.2: TP-FP approach with median filtering script diagram



Our first trials produced a high number of TP values compared to FP values which is not possible when we consider the performance results of the model. This is caused by the methodology used to create the RUL values using SQL cleanup script. The script starts marking RUL values zero at the instance of stop and continues marking the following lines zero until the end of downtime. This produces repeating zero values in the RUL column which is used to calculate the TP values.

After completing the prediction, the code necessary for TP-FP calculation is processing as follows from the beginning of predicted data, also shown in the Figure 4.2:

i. Calculate the median of "n" number of rows.
ii. If the median value of the "n" number of rows is between 0 and 4h15m, check if there is a stop in the next 4h duration in the original dataset (starting from nth row).
    a. If so, mark TP and return to *ii* for the next "n" number of rows in prediction.
    b. If not, mark FP and return to *ii* for the next "n" number of rows in prediction.
iii. If not, return to *ii* and continue to the next n number of rows.

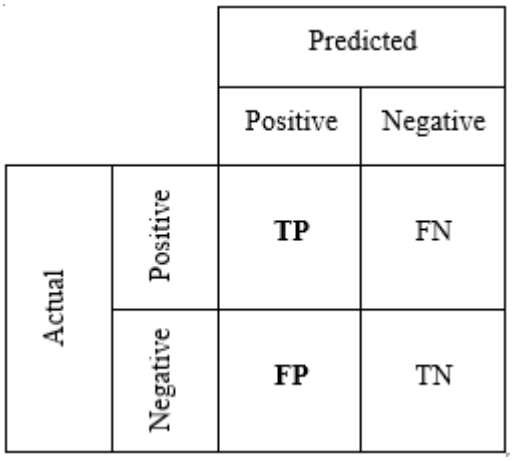
The median sizes chosen are 50, 100, 150 and 200 rows. Each line represents approximately 3 seconds so, at most, we would have 10 minutes of calculated median RUL values. Since the stops dataset contains stops that last a long duration, actual RUL values are lines of recursive zero values that cause the algorithm to produce a high number of TP values. In order to resolve the recursive TP values, timestamps of each TP value are stored as an array and checked for the unique timestamps of TP values. By doing so we were able to acquire the unique TP values.

As a slight improvement in the calculation process, we changed the behavior after TP detection where previously the filter was scanning the next n number of rows. With the upgrade, it calculates the median of n number of rows right after the row where TP value is detected. This final approach slightly decreased the TP and FP values but increased the accuracy of the model performance.

4.4.4 Confusion Matrix

Using confusion matrix (Figure 4.3) distribution, we were able to calculate the sensitivity, also called Recall of the applied model where proposed evaluation metrics were not capable of explaining the model performance.

Figure 4.3: Confusion matrix



In this study our focus towards a confusion matrix points the *TP* and *FP* values. *TP* values represent the scenario where prediction outputs and error and within the estimated time if there is an actual error. On the contrary *FP* represents output of error where there is actually no error. Because we have the dataset of past, we already know the total number of stops, and in our case positive values (*P*). Using these three variables we were able to calculate the Precision (Equation 4.6) and Sensitivity (Equation 4.7) of the prediction model.

$$Precision = \frac{TP}{TP+FP} \quad \textbf{(4.6)}$$

$$Sensitivity = \frac{TP}{P} \quad \textbf{(4.7)}$$

5. RESULTS

5.1 TEST SET

5.1.1 Standard Dataset

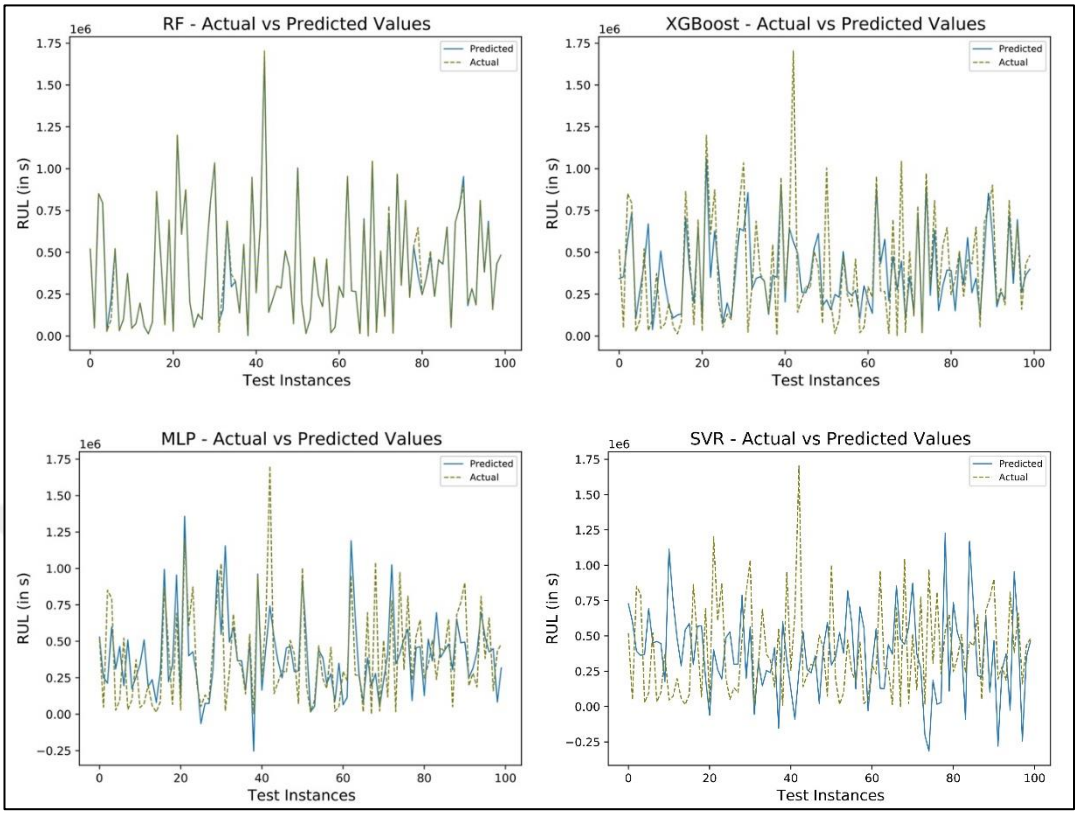
After completing the model fitting processes and determining the best fit parameters for each model, created models are tested for the test dataset. Evaluation matrices and the results of each model are shown in the figure. As a model evaluation technique, inverse MAPE is calculated by subtracting the result from 100, where it is used to explain the performance of the model by obtaining an error free percentage.

Table 5.1: Performance results of standard dataset

Algorithm	R^2	MAE	MAPE	Inverse MAPE	RMSE
RF	0,990	8855,15	2.09	97.91	45316.91
XGB	0,650	173871,26	41.04	58.96	272467.00
SVR	-0,435	407963.56	96.3	3.7	551882.41
MLP	0,450	234300.15	55.31	44.69	341651.11

Best results were obtained using the RF model, which seem to fit data significantly and the R^2 value close to one shows how accurate the prediction is (Table 5.1). Followed by XGBoost which obtained R^2 value of 0,65. Similarly when the MAPE scores were compared, RF model outperforms the rest with a score of 2,09 percent. For detailed analysis model prediction results are plotted and observed.

Figure 5.1: Predicted vs. actual plot of RUL for standard dataset



From the actual vs. predicted plots shown in Figure 5.1 we observed how well RF model fit the model and drew a prediction line similar to actual RUL values. Although XGB and MLP obtained lover scores, those prediction plots also showed potential in prediction. Worst performance is obtained by SVR model and so it is not used in the upcoming tests.

5.1.2 Stops-Removed Dataset

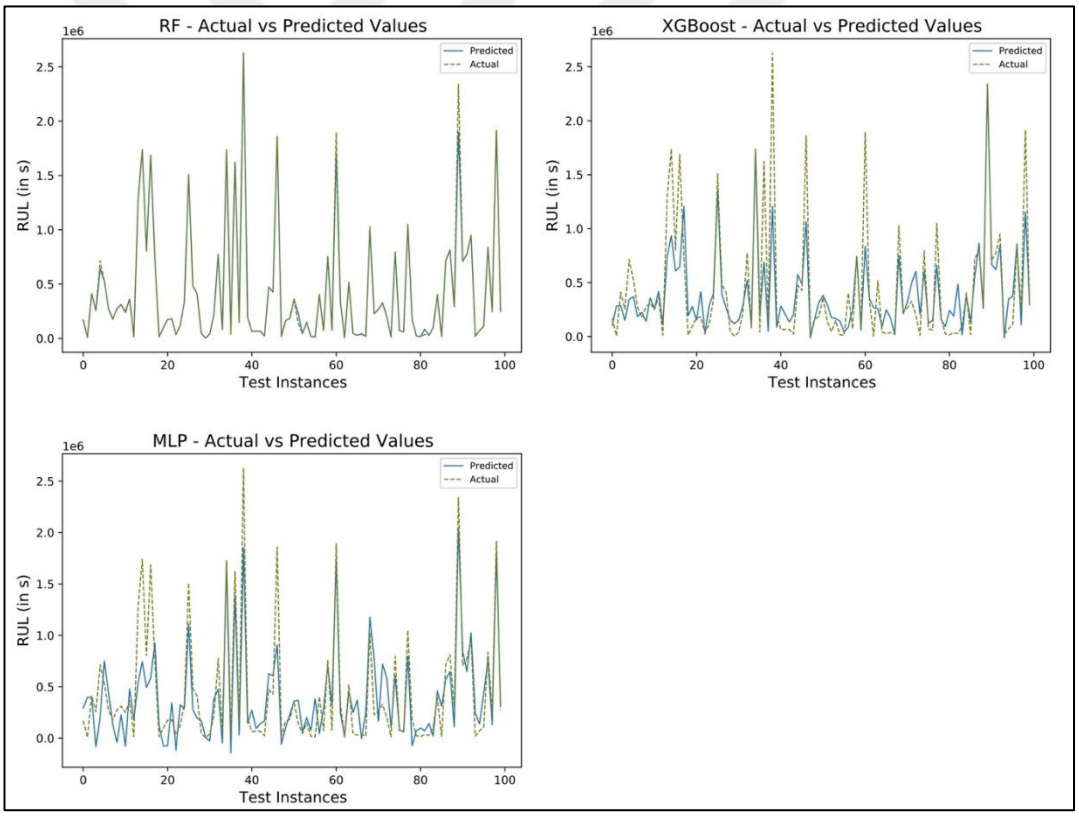
When the poor performance of MLP and SVR were examined, compared to RF and XGB, Stops-Removed datasets were used to create and predict model results for comparison. Since this dataset contains only the instance of stops rather than whole stop duration, filtered information could increase the performance of the prediction model. This detail is addressed after investigation the sensor readings during stops where it is observed that while some sensors do not change reading ranges during stops, others showed change in observations. And so this can result in failure of a promising model fit for prediction. The

results for the Stops-Removed dataset are shown in the Table 5.2 where improvements in all models are observed which supports the approach for the filtered dataset.

Table 5.2: Performance results of stops-removed dataset

Algorithm	R^2	MAE	MAPE	Inverse MAPE	RMSE
RF	0.994	4895.14	1.12	98.88	37567.86
XGB	0.744	167505.2	38.49	61.51	253658.63
MLP	0.695	192165.52	44.15	55.85	276604.36

Figure 5.2: Predicted vs. actual plot of RUL for stops-removed dataset



5.1.3 Correlations Removed Dataset

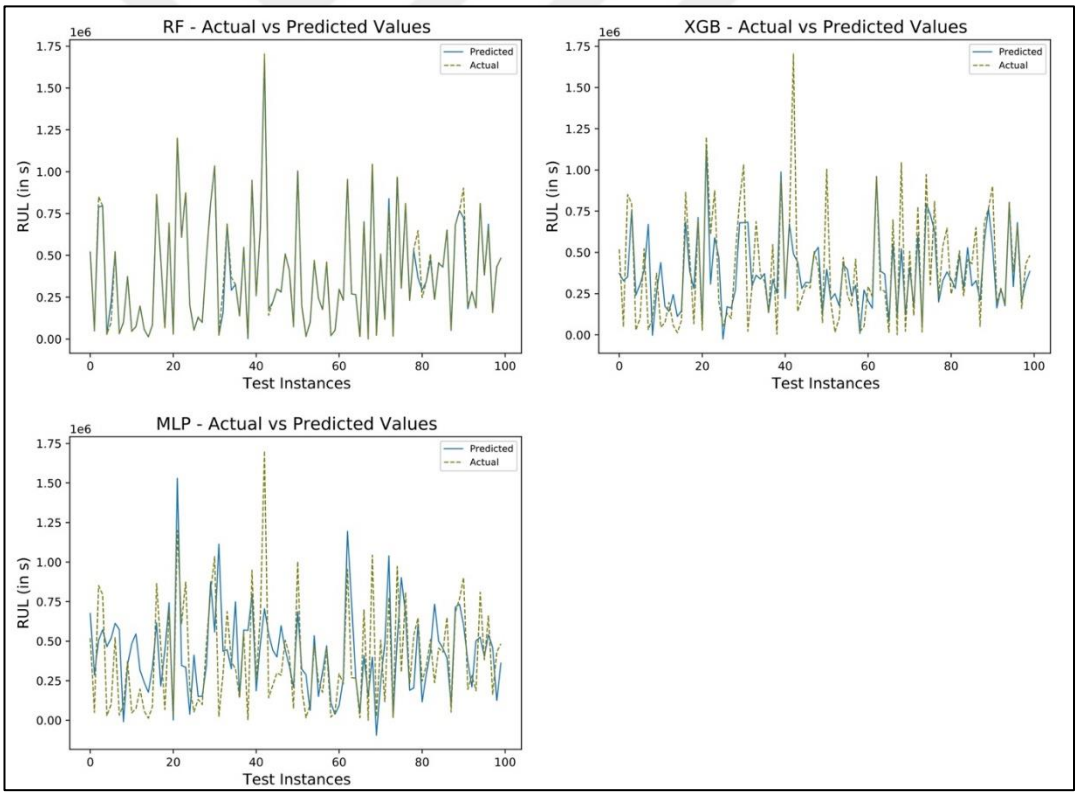
Our third approach for the test set continued with the use of correlations removed dataset where it is tested and the results for the following dataset is shown below.

Table 5.3: Performance results of correlations removed dataset

Algorithm	R^2	MAE	MAPE	Inverse MAPE	RMSE
RF	0.99	9093.31	2.15	97.85	46234.43
XGB	0.626	181337.89	42.81	57.19	281839.40
MLP	0.467	235424.8	55.57	44.43	336252.33

Compared to previous models, a slight decrease in prediction performance is observed. Similar to our first two trials, RF model fit the test set significantly. On the other hand, XGB obtained the lowest score so far by using correlations removed dataset.

Figure 5.3: Predicted vs. actual plot of RUL for correlations removed dataset



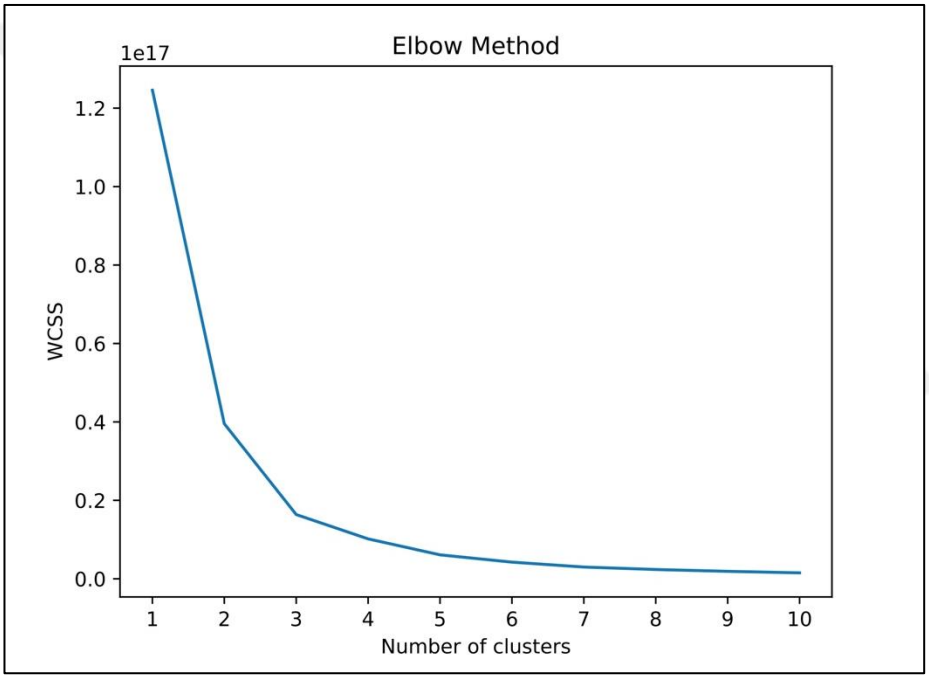
5.1.4 Stops-Removed and Clustering Applied Dataset

Our final approach is the use of clustering application. The aim of this approach is to add an external information to the dataset using the internal dynamics. We tried to observe

the cluster distributions and add the cluster label as a new feature to the dataset. The use of clusters in literature displays classification approaches for maintenance purpose as well. Because our problem relies on regression and predicting RUL in time domain, a meaningful class distribution could lead to increase in prediction.

By calculating WCSS score and using elbow method, we are able to determine the number of optimal clusters available in the dataset distribution. The figure shows the number of clusters tried during observations and their WCSS scores.

Figure 5.4: Elbow method plot of WCSS vs. number of clusters



When the original dataset is used for clustering, resulting graph did not provide a clean elbow to observe. The reason behind creating stops-removed dataset mentioned in chapter four can explain this result. For this matter clustering, stops-removed dataset is used and shown in figure, there are three meaningful clusters.

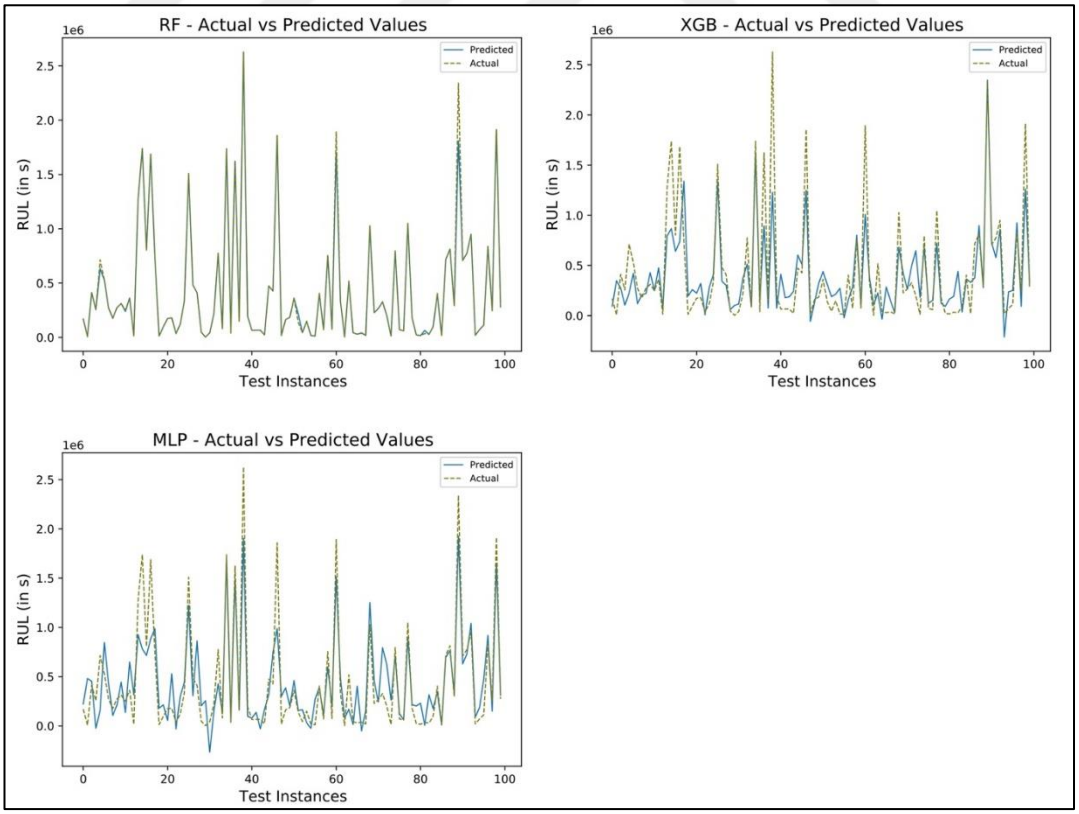
Created cluster labels are added to the dataset as a new feature and the model is created using this extra attribute. In the meantime, the cluster model is also stored and for the final test, label values are added to each new line and predictions are made using this new feature as well.

Table 5.4: Performance results of stops-removed and clustering applied dataset

Algorithm	R^2	MAE	MAPE	Inverse MAPE	RMSE
RF	0.994	4954.07	1.14	98.86	38118.35
XGB	0.748	165838.26	38.1	61.9	251425.99
MLP	0.709	186614.23	42.88	57.12	270425.70

The results for clustering applied dataset is very promising where an increase is observed in each model. This proved our approach for clustering. When the plots for predictions vs actual data is checked, both XGB and MLP obtained better fit to actual values compared to previous approaches while RF kept its performance almost identical with a slight improvement.

Figure 5.5: Predicted vs. actual plot of RUL for stops-removed and clustering applied dataset



5.2 NEW VALIDATION SET

After completing evaluations on current dataset, we were able to obtain a new dataset form the following month and used it for testing the results for above mentioned models. New dataset is processed the same way explained in chapter three.

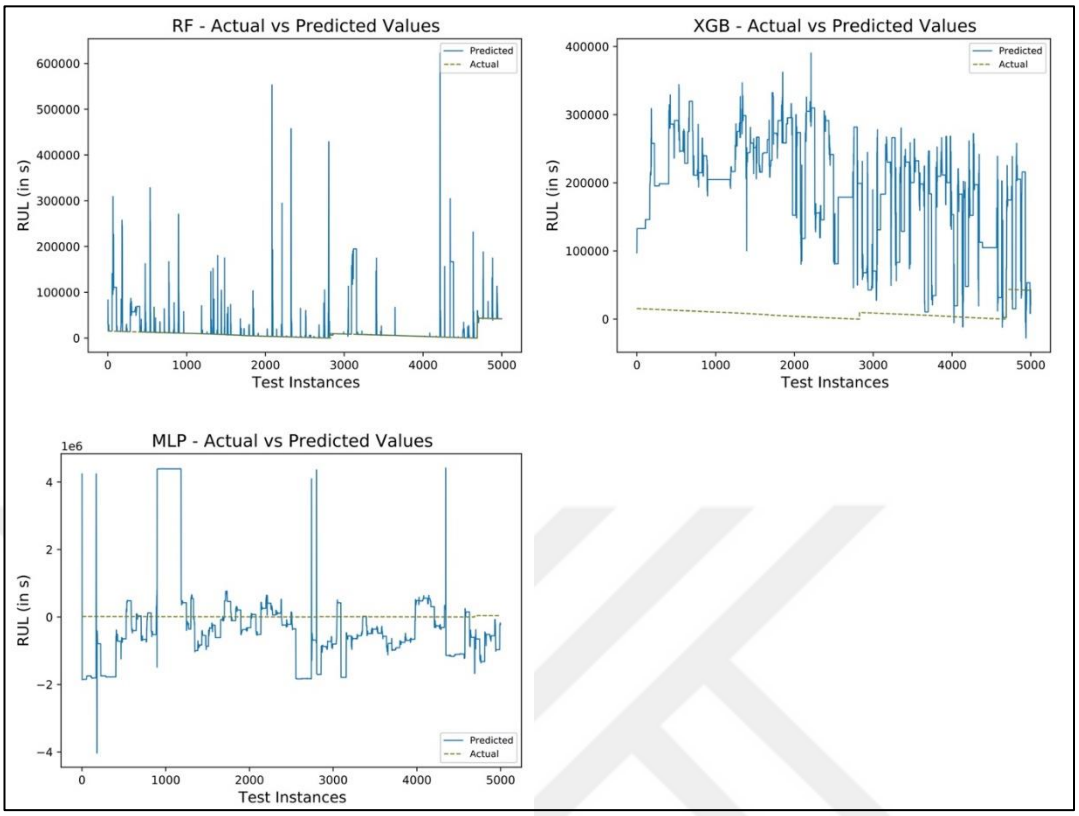
The results for standard dataset were unexpected. Model performance results pointed how poorly each model performed on prediction.

Table 5.5: Performance results of newly generated validation set

Algorithm	R^2	MAE	MAPE	Inverse MAPE	RMSE
RF	-2.15	210479.27	121.56	-21.56	286963.58
XGB	-0.295	140958.46	81.41	18.59	183876.99
MLP	-222.02	2079551.73	1200.98	-1100.98	2413377.23

These results required detailed monitoring of the predictions in order to find a meaningful explanation. After the plots are observed, we were able to see how the model managed to fit actual values on some intervals. RF obviously obtained the best fit among other models, while trend in the prediction somehow represented the actual values in other models shown in Figure 5.6.

Figure 5.6: Predicted vs. actual plot of RUL for new validation set



While other models for the remaining three dataset approaches are checked, similar results are obtained as a result those results are omitted from this report since no additional information is obtained.

5.3 NEW EVALUATION APPROACH

Due to the unsatisfactory performance results on the new validation test, our focus shifted to the evaluation of the model performances.

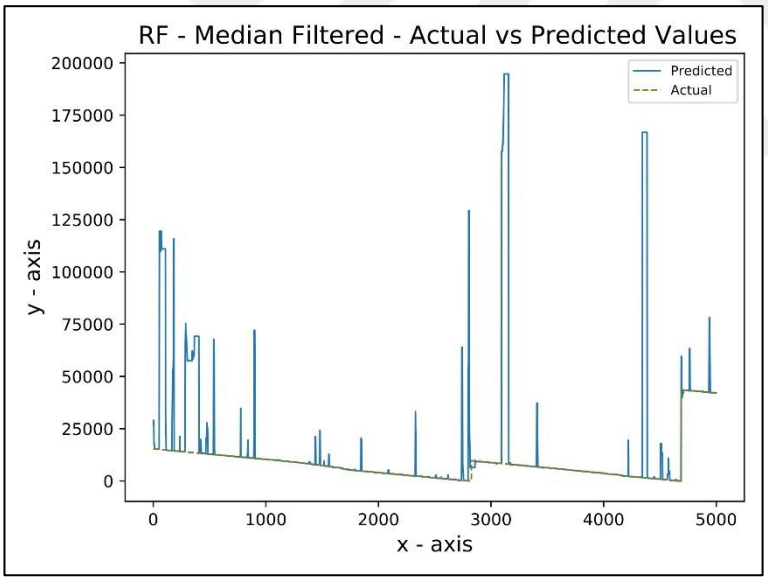
5.3.1 Median Filter

After completing the model performance evaluations with the training and test sets, newly generated validation set is used for prediction.

Using the preferred prediction model with the newly generated dataset of 1 month, we were able to see the model prediction performance results, also shown in the Figure 5.6. The variance of the prediction compared to the actual RUL values resulted in poor prediction results. Because the prediction results are not distributed as expected where the predicted RUL values would decrease linearly, new methods are applied. The random spikes in prediction caused the model to perform weakly. To overcome this problem our first approach became using median filtering.

After the prediction, median filtering is applied to the results to smooth the predicted RUL values as well as the general distribution. But since the predictions are already unevenly distributed, using just a median filter is not enough to increase model performance where random spikes still exists (Figure 5.7).

Figure 5.7: Median filter applied prediction results for RF model



Different parameters are used for median filter. These parameters represent the number of lines used to calculate the median and applied to the prediction results. Increasing the range for the median, resulted in smoother transaction in the RUL values but had a high effect on the data since it manipulated the results more than expected. Lower values of median parameters performed poorly. Using the results from the median filter applied RUL values, no significant change in the performance of the model is observed. For this

matter our next approach become focusing on the calculation of True-Positive and False-Positive values.

5.3.2 True Positive – False Positive Approach

The fact that median filter alone did not handle the high variance in the prediction results using new validation set, our focus shifted towards defining global True and False variables. This approach is a method of filtering out false alarms while providing decent RUL predictions to the assembly line. Another advantage of this approach is creating an evaluation metric for the model performance using confusion matrix technique and calculating sensitivity. The results for the basic model are provided in the table.

Table 5.6: Validation set TP-FP results

Model	TP	FP
RF	234	667
XGB	33	3834
MLP	5	557

As observed from the results of TP-FP approach, the method required rearrangement since no meaningful outputs were obtained. Such high number on TP and FP values are inconsistent. When the algorithm is checked, the flaws in the approach are addressed.

5.3.3 True Positive – False Positive Approach with Median Filtering

As stated in the methodology, this approach is a new observation approach in our study where our focus shifted from accurate prediction of each instance to overall accuracy of the model for RUL prediction. This new approach also brings the implementation details into focus. Since the dataset used in this study is a real-world data, the approach also must provide real-time implementation methodologies.

Below shared results are obtained using RF models only. MLP and XGB models were not able to capture any promising results. When previously observed predictions for new validation set is taken into account, this is not a surprising result. Four different model creation approaches are used for this new methodology to observe model performance results. Each model is presented on their own tables.

Table 5.7: Main dataset RF model prediction results on validation set

RUL Alarm	Median Size	TP	FP	Unique TP	Unique FP	Prec.	Sens.
3600s (1h)	50	73	1	11	1	0,917	0,379
	100	36	1	10	1	0,909	0,345
	150	20	1	7	1	0,875	0,241
	200	19	0	8	0	1,000	0,276
5400s (1,5h)	50	113	3	12	3	0,800	0,414
	100	58	2	10	2	0,833	0,345
	150	34	2	7	2	0,778	0,241
	200	28	1	8	1	0,889	0,276
7200s (2h)	50	155	5	13	5	0,722	0,448
	100	79	2	10	2	0,833	0,345
	150	49	3	7	3	0,700	0,241
	200	39	1	8	1	0,889	0,276
10800s (3h)	50	222	5	14	5	0,737	0,483
	100	113	3	11	3	0,786	0,379
	150	71	2	7	2	0,778	0,241
	200	57	1	9	1	0,900	0,310

Table 5.8: Stops removed dataset RF model prediction results on validation set

RUL Alarm	Median Size	TP	FP	Unique TP	Unique FP	Prec.	Sens.
3600s (1h)	50	47	1	12	1	0,923	0,414
	100	21	1	8	1	0,889	0,276
	150	13	0	6	0	**1,000**	0,207
	200	11	0	5	0	**1,000**	0,172
5400s (1,5h)	50	84	1	12	1	0,923	0,414
	100	40	1	8	1	0,889	0,276
	150	25	0	7	0	1,000	0,241
	200	18	0	6	0	1,000	0,207
7200s (2h)	50	123	2	12	2	0,857	0,414
	100	58	1	8	1	0,889	0,276
	150	39	0	8	0	1,000	0,276
	200	28	0	7	0	1,000	0,241
10800s (3h)	50	182	4	13	4	0,765	0,448
	100	89	1	9	1	0,900	0,310
	150	58	1	8	1	0,889	0,276
	200	43	0	7	0	1,000	0,241

Table 5.9: Correlations removed dataset RF model prediction results on validation set

RUL Alarm	Median Size	TP	FP	Unique TP	Unique FP	Prec.	Sens.
3600s (1h)	50	75	2	13	2	0,867	0,448
	100	36	1	9	1	0,900	0,310
	150	22	2	8	2	0,800	0,276
	200	19	1	8	1	0,889	0,276
5400s (1,5h)	50	116	2	13	2	0,867	0,448
	100	57	1	9	1	0,900	0,310
	150	36	2	8	2	0,800	0,276
	200	27	1	8	1	0,889	0,276
7200s (2h)	50	154	4	13	4	0,765	0,448
	100	75	1	9	1	0,900	0,310
	150	50	3	8	3	0,727	0,276
	200	37	1	8	1	0,889	0,276
10800s (3h)	50	224	10	14	10	0,583	0,483
	100	111	5	11	5	0,688	0,379
	150	72	4	8	4	0,667	0,276
	200	54	3	8	3	0,727	0,276

Table 5.10: Cluster added dataset RF model prediction results on validation set

RUL Alarm	Median Size	TP	FP	Unique TP	Unique FP	Prec.	Sens.
3600s (1h)	50	24	19	3	19	0,136	0,103
	100	11	9	2	9	0,182	0,069
	150	7	7	2	7	0,222	0,069
	200	5	5	2	5	0,286	0,069
5400s (1,5h)	50	39	40	3	39	0,071	0,103
	100	19	20	3	20	0,130	0,103
	150	13	13	3	13	0,188	0,103
	200	9	9	2	9	0,182	0,069
7200s (2h)	50	58	59	3	58	0,049	0,103
	100	28	30	3	30	0,091	0,103
	150	18	21	4	21	0,160	0,138
	200	14	14	3	14	0,176	0,103
10800s (3h)	50	92	86	3	84	0,034	0,103
	100	46	41	3	41	0,068	0,103
	150	31	29	4	29	0,121	0,138
	200	22	21	3	21	0,125	0,103

Tables show four different windows sizes of median values used to calculate TP-FP values. Each model is tested for four different RUL values as a time interval. These values represent the actual behavior of the model running at the production line. The minimum time gap chosen for model alarm is 1 hour and maximum is 3 hour. Unique values represent the amount of unique stops detected from the original dataset. The dataset tested above contains 29 unplanned stops.

At first glance it is observed how Stops Removed dataset with cluster labels added performed poorly, detecting four TP values as best prediction while producing high number of FP values. Using the confusion matrix metrics and calculating Precision and Sensitivity of all results, it is observed that RF model generated from stops-removed dataset predictions outperformed the remaining alternatives. Best results are obtained using median size of 50. As for time interval, 1h and 1,5h results are similar to each other, produced the most accurate predictions.

6. DISCUSSION

With the real-life dataset used in this study, our first focus was processing such data. The first decision made was how the prediction was going to be made. Choosing a classification approach would not be feasible in a real-life scenario where the model would make predictions based on an instance, for that instance. In order to obtain a preferable model, we focused on the regression task and predict the RUL in the domain of time.

For this purpose, four different datasets were created and used in the experiments conducted on chapter five. Even though the most of the surveyed literature have focused on time series data analysis, in our case that is not an applicable approach. Due to the fact that it is not possible for the model to verify the actual time to failure at the run time and computational costs, four different machine learning algorithms namely RF, XGB, MLP and SVR were selected and trained for predicting potential failures ahead of time.

Model prediction results pointed out the performance of RF algorithm where in each case it produced the highest R^2 value. This is due to its complex nature, creating multiple decision trees, in our case resulted in a markable fit of the model.

With the introduction of the newly generated dataset, we took the opportunity to test our models on the new validation set. The results were unexpected and unsatisfactory. To understand the problem, we further investigated the prediction results. Although looking at these results it can be said that there is an overfitting problem in the models, we do not think that is the case. When the prediction plots for new validation set is analyzed, notable model fit was observed for RF models. The fact that our RUL creation method produces label values that linearly decrease until the next error date, in real-life that is not the case. Actual errors that occur in the production line are sometimes caused by an instance of a material feed and has nothing to do with the component error. When the problem is handled in this manner, it can be said that our previously trained models overlearned the data, but as well as error situations.

The reason behind unsuccessful results from RF model can be explained with the high amount of variance in the prediction results which can be observed from the plots. Considering the model fit on the plot, our focus is shifted towards valuation of the prediction results, and we introduced a new layer in the RUL prediction namely TP-FP approach.

This approach has two major aspects. First, we try to predict the RUL of assembly line by providing relevant results for necessary actions to be taken. It is observed that, standalone ML algorithms is not sufficient for such real-world production dataset where unpredictable effects are present in the dataset compared to a synthetic dataset. Second, which is the most important finding of this study, providing just a prediction model is not sufficient where human-machine interaction and communication is vital. Our new approach can be observed as a framework definition where we try to minimize the effects caused by the real-world system and provide an information technique based on ML results.

Our first approach resulted in incoherent values since such number of true or false alarms makes no sense. The problem with the first approach was caused by the filtering method used where we scanned the dataset row by row after each FP values obtained. Even though this detail is addressed and fixed, yet we were not able to obtain any meaningful results. Based on our observations on the predictions, the variation on the prediction distribution directed the approach to filtering techniques but standalone filtering applied models were not enough to obtain promising results.

Our final approach become combining these two methods explained above and creating a final evaluation method. Different from previous TP-FP approach, in this new method we introduced median window sizes. While we were not able to obtain any promising results from the median filter alone, we decided on using the median filter for prediction result smoothing before TP-FP check. This new method showed promising results where we were able to obtain a value of 0,923 for precision of the results.

For evaluation of the currently proposed technique, newly generated datasets should be tested to see if there is a meaningful outcome. During the literature review, approaches like confusion matrix classification for RUL prediction are observed (W. J. Lee et al. 2019), and as a future work these approaches can be tried. As a second bullet point, clustering methods can be tested for RUL prediction where repetitively predicted so called “failure” instances can be used as an alarm situation. Finally, computationally more complex models can be tried for comparison to the present study.

7. CONCLUSION

With continuous advancements in machine learning, more and more industry 4.0 applications are expected to emerge. Due to the integration of AI in the industry, previously applied human-machine interactions began to shift its characteristic. Maintenance approaches in the past either resulted in early replacement of machine parts or disruptions in production lines due to failures. What these scenarios produce is either waste of material or waste of time and as a result waste of money.

In this study, it is aimed to propose a prediction model for the real-life production line. Four different ML algorithms namely, RF, XGB, MLP and SVR were used to create models with different approaches to the existing dataset. It is very important to note that the problem for ML application starts at data cleaning and preprocessing procedures. We proposed four different datasets using the real-life data stored in the manufacturing line and compared the results of all possible variations. Among all proposed methods RF, an ensemble bagging method, proved to perform best with our dataset, followed by XGB.

Although notable results were obtained from ML algorithms, with the introduction of newly received dataset, unsatisfactory results were obtained using previously created models. These results shifted our approach towards defining a new evaluation method. With this new approach we tried to filter the results from the prediction model and eliminate the effects of high variation in the prediction results. This added layer to the prediction model showed reasonable results, predicting around 42 percent of the errors occurred in the production line based on the dataset.

In future work, we plan on testing our proposed method on newly generated dataset for comparison purposes and use different methods for RUL prediction. For instance, different than current regression approach, clustering approaches can be implemented and used to predict RUL of the production line.

REFERENCES

Periodicals

Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*, 114598. https://doi.org/10.1016/j.eswa.2021.114598

Canziani, A., Paszke, A., & Culurciello, E. (2017). An Analysis of Deep Neural Network Models for Practical Applications. *ArXiv:1605.07678 [Cs]*. http://arxiv.org/abs/1605.07678

Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S., & Syntetos, A. A. (2020). Predictive maintenance using cox proportional hazard deep learning. *Advanced Engineering Informatics*, *44*, 101054. https://doi.org/10.1016/j.aei.2020.101054

Ho, S. L., Xie, M., & Goh, T. N. (2002). A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers & Industrial Engineering*, *42*(2–4), 371–375. https://doi.org/10.1016/S0360-8352(02)00036-0

Jasiulewicz-Kaczmarek, M., Legutko, S., & Kluk, P. (2020). *Management and Production Engineering Review*. https://doi.org/10.24425/MPER.2020.133730

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and Understanding Recurrent Networks. *ArXiv:1506.02078 [Cs]*. http://arxiv.org/abs/1506.02078

Lee, J., Davari, H., Singh, J., & Pandhare, V. (2018). Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *18*, 20–23. https://doi.org/10.1016/j.mfglet.2018.09.002

Lee, W. J., Wu, H., Yun, H., Kim, H., Jun, M. B. G., & Sutherland, J. W. (2019). Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data. *Procedia CIRP*, *80*, 506–511. https://doi.org/10.1016/j.procir.2018.12.019

Li, Z., Wang, Y., & Wang, K.-S. (2017). Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. *Advances in Manufacturing*, *5*(4), 377–387. https://doi.org/10.1007/s40436-017-0203-8

Machado, C. G., Winroth, M. P., & Ribeiro da Silva, E. H. D. (2020). Sustainable manufacturing in Industry 4.0: An emerging research agenda. *International Journal of Production Research*, *58*(5), 1462–1484. https://doi.org/10.1080/00207543.2019.1652777

Mocanu, E., Nguyen, P. H., Gibescu, M., & Kling, W. L. (2016). Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, *6*, 91–99. https://doi.org/10.1016/j.segan.2016.02.005

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *ArXiv:1712.04301 [Cs]*. http://arxiv.org/abs/1712.04301

Rieger, T., Regier, S., Stengel, I., & Clarke, N. (2019). Fast Predictive Maintenance in Industrial Internet of Things (IIoT) with Deep Learning (DL): A Review. *Internet of Things*, 11.

Rivera, D. L., Scholz, M. R., Fritscher, M., Krauss, M., & Schilling, K. (2018). Towards a Predictive Maintenance System of a Hydraulic Pump. *IFAC-PapersOnLine*, *51*(11), 447–452. https://doi.org/10.1016/j.ifacol.2018.08.346

Traini, E., Bruno, G., D'Antonio, G., & Lombardi, F. (2019). Machine Learning Framework for Predictive Maintenance in Milling. *IFAC-PapersOnLine*, *52*(13), 177–182. https://doi.org/10.1016/j.ifacol.2019.11.172

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, *101*, 158–168. https://doi.org/10.1016/j.comnet.2015.12.017

Wright, S. A., & Schultz, A. E. (2018). The rising tide of artificial intelligence and business automation: Developing an ethical framework. *Business Horizons*, *61*(6), 823–832. https://doi.org/10.1016/j.bushor.2018.07.001

Xie, X., Wu, D., Liu, S., & Li, R. (2021). IoT Data Analytics Using Deep Learning. *ArXiv:1708.03854 [Cs]*. http://arxiv.org/abs/1708.03854

Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018a). Deep Learning for Improved System Remaining Life Prediction. *Procedia CIRP*, *72*, 1033–1038. https://doi.org/10.1016/j.procir.2018.03.262

Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018b). Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems*, *48*, 78–86. https://doi.org/10.1016/j.jmsy.2018.05.011

Other Publications

Francis, F., & Mohan, M. (2019). ARIMA Model based Real Time Trend Analysis for Predictive Maintenance. *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 735–739. https://doi.org/10.1109/ICECA.2019.8822191

Gensler, A., Henze, J., Sick, B., & Raabe, N. (2016). Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 002858–002865. https://doi.org/10.1109/SMC.2016.7844673

Kanawaday, A., & Sane, A. (2017). Machine learning for predictive maintenance of industrial machines using IoT sensor data. *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 87–90. https://doi.org/10.1109/ICSESS.2017.8342870

Shimada, J., & Sakajo, S. (2016). A statistical approach to reduce failure facilities based on predictive maintenance. *2016 International Joint Conference on Neural Networks (IJCNN)*, 5156–5160. https://doi.org/10.1109/IJCNN.2016.7727880

Song, Z., & Moon, Y. (2018). CyberManufacturing System: A Solution for Sustainable Manufacturing. *Volume 2: Advanced Manufacturing*, V002T02A068. https://doi.org/10.1115/IMECE2018-86092