Data-Driven Prediction of Ship Propulsion Power Using Spark Parallel Random Forest on Comprehensive Ship Operation Data

Qin. Liang, Erik. Vanem, Knut E. Knutsen, Houxiang. Zhang

Abstract— **This paper aims to propose an efficient machine learning framework for maritime big data and use it to train a random forest model to estimate ships' propulsion power based on ship operation data. The comprehensive data include dynamic operations, ship characteristics and environment. The details of data processing, model configuration, training and performance benchmarking will be introduced. Both scikit-learn and Spark MLlib were used in the process to find the best configuration of hyperparameters. With this combination, the search and training are much more efficient and can be executed on latest cloud-based solutions. The result shows random forest is a feasible and robust method for ship propulsion power prediction on large datasets. The best performing model achieved a R2 score of 0.9238.**

I. INTRODUCTION

The International Maritime Organization (IMO) continues to push for the reduction of Green House Gas (GHG) emissions from ships. The environmental requirements for ships are becoming stricter. From the latest discussion within the Marine Environment Protection Committee (MEPC) 77 [1], the Committee recognized the need to further strengthen the ambition to cut the GHG emissions from ships. As of today, the current strategy requires international shipping to reduce CO2 emissions by at least 40% by 2030, 70% by 2050 compared to 2008 and to peak the GHG emissions from international shipping as soon as possible. In order to achieve this, it is crucial to have an appropriate and accurate method to evaluate the performance of international shipping. The Maritime industry is, like other industries, subject to an ongoing digital transformation. As a result, increased data availability and advanced analytics create opportunities for novel data-driven services, for example for real-time performance monitoring of ocean-going ships. Ship propulsion power is one of the most important parameters of ship operational performance. If the propulsion power is known, other performance related parameters, e.g., engine load factor, fuel consumption and different kinds of emissions could be calculated from it. Traditional techniques for ship propulsion power prediction rely on analysis of calm water resistance, added resistance due to environment or on towing tank tests at model scale [2]. It has advantages but is time consuming and requires many parameters to do the evaluation. To implement these methods on a fleet can be extremely difficult and might not be possible to achieve good accuracy. On the other hand, simplified methods based the relationship between speed and power can be applied to the fleet, but it ignores the fluctuations for weather and have many uncertainties. A model which can monitor the performance of global fleet accurately and efficiently becomes demanding and important to monitor and tackle the reduction of GHG emissions.

In this paper, an efficient random forest (RF) model for ship propulsion power prediction is proposed. Automatic grid search was adopted in the training process to find the best performed model in an optimized manner. Normally, scikit-learn will be used for different machine learning studies. Considering compatibility with big data and requirements for parallel computing, Spark MLlib was also used in this study. Based on the evaluation of the performance, RF was found to be a viable, robust technique and scalable to big data like global fleet.

The comprehensive ship operation data used in this paper include Automatic Identification System (AIS), IHS Fairplay, ECMWF (European Centre for Medium-Range Weather Forecasts) and onboard performance monitoring data. AIS is a GPS-based ship tracking system required on all internationally trading ships with 300 or more gross tonnage (GT) or passenger ships of any size. It can monitor and track most of the international shipping activities, and provides data on position, heading, speed, etc. The IHS Fairplay is one of the largest maritime databases which cover ship characteristics and technical information. The ECMWF data include global weather and sea state information from satellite and available observations. The onboard performance monitoring data include verified noon reports and torque meter measurements. In previous related studies [3][4] various machine learning methods and artificial neural networks (ANN) for ship propulsion power prediction have been explored thoroughly [2, 3]. In these previous studies, Spark was used to process the data because of its speed and capability to handle huge amounts of data by parallelization. For the machine learning part, however, a single node implementation was adopted. In this paper, the usage of Apache Spark was extended for both data processing and analytics.

Apache Spark a popular big data processing framework. More end users start to transfer on-premises databases to cloud-based data warehouses or data lakes. From these transitions, efficient and accurate data-driven models which can be executed on these modern platforms should be explored.

ANN models have been used in a number of different applications due to their strong ability to summarize insights from complex and abstract data. However, several limitations of ANNs are well known. First, it requires a huge amount of training data with good quality. This can be difficult to achieve from real-world scenarios. For example, the monitoring data in this study require human efforts and installation of logging devices onboard. Second, ANNs need expensive hardware and

Manuscript 202 submitted to 2022 IEEE 17th International Conference on Control & Automation (ICCA). Received February 22, 2022.

much longer training time compared to many other machine learning. Lastly, the trained network is usually treated as a black box, where it is almost impossible to know the meaning of the trained parameters in the network. ANNs are capable of handling a variety of data problems, but for certain problems, other machine learning models may fit better than ANNs. That is one of the reasons why RF was chosen. The other reasons, details of data processing, model training and evaluation of different RF will be introduced and discussed. In the end, a machine learning life cycle management framework based on the trained RF model for maritime big data will be proposed. It can be used as a reference for other maritime data applications

II. METHODS

A. Random forest

Random forest is a kind of supervised machine learning algorithm based on ensemble learning. It was proposed by Leo Breiman in 2001 [5] . RF has been extremely successful as a general-purpose classification and regression method [6]. Ensemble learning means several types of algorithms, or several instances of the same algorithm are combined multiple times to form a more powerful model. For RF, numerous randomized decision trees will be created on each subsample of the dataset, then the outputs will be aggregated to improve prediction accuracy and avoid overfitting. After iterative searching, a 'randomized decision tree' model will be trained.

RF is a popular ML algorithm that has been used widely for different prediction problems in the maritime domain. For example, Budonov et al. [7] developed a model with a combination of different RF and decision tree methods for ship destination prediction with an accuracy of 97%. Zhong et al. [8] trained a RF model based on AIS data for vessel classification. The model can be used to classify three major ship types from geometric features. These two examples used RF for classification, it can also be used for regression when a continuous value is predicted.

There are several outstanding features being the reasons that RF was selected in this study. With RF, the complex relationships of the input features with each other can be modelled and considered relatively robust with regards to outliers. Many studies have shown that RF has a high level of predictive accuracy even with the presence of noise, outliers, and with regards to overfitting. Compared to other ML algorithms (e.g. ANN and Support Vector Regression), RF has fewer hyperparameters that need to be tuned which reduce the subjectivity of training. Because of the characteristics of its own structure, RF does not need to normalize the input data. This reduces the possibility of potential error due to data with different scales. In addition, random forest provides convenience with handling of gaps in the data. Hence, RF as an efficient and robust algorithm was selected in this study.

B. Spark parallel data processing and MLlib

The transition towards digitalization and automation is speeding up in the maritime industry. Digital technologies are being used to increase competitiveness and enhance operational efficiency. With the development of Internet of Things (IoT), sensors, communication between ship and shore, onboard edge computing and storage, more data become available. Huge amounts of data are being transferred and stored. To process and analyze these big data efficiently become essential and crucial. There is a certain number of distributed data processing frameworks. The high pace of development leads to some of these becoming outdated and gradually disappearing from the community, while others become more popular, like Apache Spark. The core principle of big data processing is parallel processing with a cluster of computing nodes. Compared with a single node, there is no bottleneck on individual performance. The Spark master node can scale up the cluster automatically based on user's configuration and requirements. In this study, Azure Databricks from Microsoft was used as Spark environment and data processing platform. Databricks is a unified data analytics platform also a company founded by the original creators of Apache Spark.

Scikit-learn is one of the most popular machine learning libraries in Python. It provides various classification regression and clustering algorithms including support vector machines, random forest, gradient boosting and k-means [11]. Scikit-learn uses in-memory processing and provide very good performance if the data can be fit into memory. If the data are too large for in-memory processing, either it is not possible to train models, or it requires complex data operations to execute.

Apache Spark's Machine Learning Library (MLlib) is designed for simplicity, scalability, and easy integration with other data processing tools [12]. It is built on top of Spark and integrates seamlessly with other Spark components, etc., Spark SQL, Spark Dataframe and Spark Streaming. MLlib is a scalable machine learning library consisting of common learning algorithms and utilities like scikit-learn, including classification, regression, clustering, collaborative filtering, dimensionality reduction, and underlying optimization primitives [13] .

C. Machine learning steps

To create an efficient machine learning cycle in this study, both scikit-learn and MLlib were used. Data exploration and visualization is an important step before the implementation of machine learning. Because of Spark's lazy evaluation principle (no actual expectation until a check operation), it is difficult to explore the data. On the contrary, scikit-learn supports Pandas and Matplotlib which makes the exploration and test process very efficient.

In this study, scikit-learn was used in the first stage to test and explore appropriate configuration for the random forest model. Then the summarized configuration was applied to the major scope of machine learning with Spark MLlib. The detailed machine learning flow is illustrated in Fig. 1.

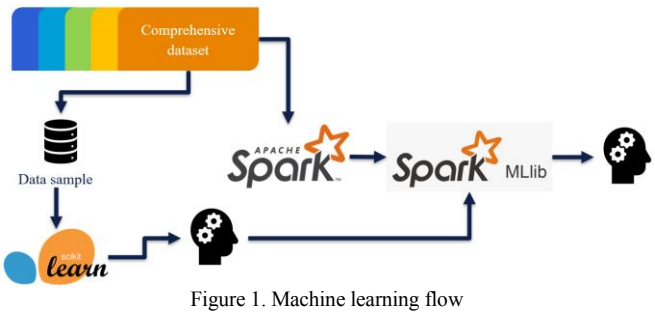

Figure 1. Machine learning flow

III. Comprehensive Ship Operation Data

A. Composition of the data

The comprehensive ship operation data includes data from different data sources with different sampling rates. For example, AIS provides basic information with position updates with a varying sample rate from 3s to 3 minutes [14]. The sampling frequency of the ECMWF weather data is once per hour with a resolution of 9 km. The performance monitoring data have different sampling rates from its two reporting sources. The data from noon reports are daily, and the data from onboard measurement are every 15 minutes. The IHS Fairplay data save ships' technical information which does not update frequently.

The dataset includes a total number of 231 ships in different size categories, ranges from 140 m to 400 m. In Fig. 2 the specific number of ships in each size category is illustrated. The well-distributed size categories suggest the compatibility of the trained model so that it can be implemented in the larger scope of a bigger fleet of ships.

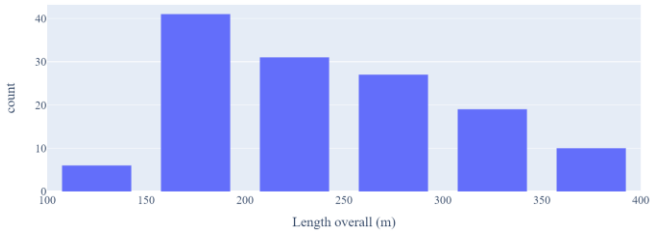

Figure 2. Distribution of length overall

B. Synchronization of data

The fragmented data from different sources cannot be directly used for machine learning. It is essential to implement appropriate data processing and synchronization. The AIS data can be joined with IHS Fairplay data with primary and foreign keys. Then this dataset can be joined with wave and wind data. The details of this synchronization operation can be found in [3]. As the final step, this dataset can be joined with the operation data based on IMO and timestamp. Fig. 3 shows the data processing flow.

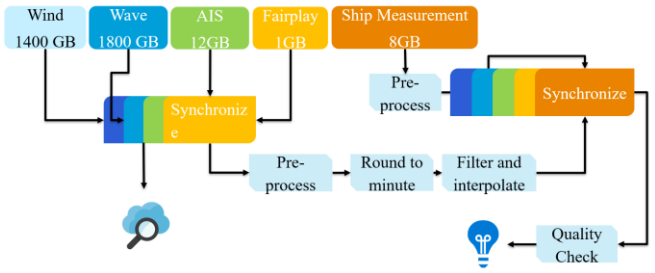

Figure 3. Data processing flow

C. Data quality and features

Machine learning models are powerful only if they are trained properly on data with high quality. Due to the increasing complexity of systems, a variety of factors can influence the data quality through the data flow from sensors to the final data storage. There is a certain number of types of data quality issues, e.g., noise, outliers, missing or duplicate records, and bad schema. Several data quality checks, corrections and filters were adopted to the synchronized comprehensive dataset.

- Data quality with heading, course over ground, wave direction and wind direction. The range of these values should be between 0 and 360. Due to different logging principles, values can be over the desired range. These values will be converted to the desired 0 to 360. In addition, the direction of the wind and wave has been altered to be relative to the direction of the ship instead of global coordinates.
- Propulsion power out of range. Ship cannot have a propulsion power that is higher than its installed power. If this kind of error is detected, the operation speed will be checked to make sure both values are in the desired range. If the operation speed is qualified, the propulsion power will be corrected to the installed power. Otherwise, this record will be erased.
- Operational speed out of range. A ship cannot have an operational speed higher than its design speed. The operational speed is based on the distance between two reporting points. To avoid filtering away too much data, a 20% margin was added to the design speed as the threshold for when operational speed is out of range.
- In port or at anchor. The model aims to predict the operational propulsion power. There are many sampling points recorded when the ships are in port or at anchor. Many uncertainties exist in these data. Therefore, data with operation speed lower than 1.5 knots were filtered out.
- Non-steady conditions. Data generated during ship acceleration and deceleration should be filtered out. For example, normal operational speed with low propulsion power (deceleration).

Table 1 includes all the features used in this study. There are more features available, these selected features are based on experience from previous studies [4]. These features have four categories: operation related, ship characteristics related, machinery and operation environment related.

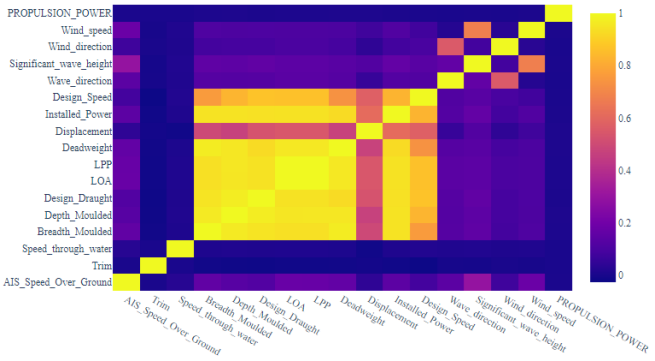

Figure 4. Correlation before data quality check

TABLE I. FEATURES FOR TRAINING

Feature category	Features
Operation related	Speed over ground
	Course over ground
	Heading
	Trim
	Speed through water
Ship characteristics related	Breadth moulded
	Depth moulded
	Design draught
	Length overall
	Length between perpendiculars
	Deadweight
	Displacement
Ship machinery related	Installed propulsion power
	Design speed
Operation environment related	Significant wave height
	Mean wave direction
	Wind speed
	Mean wind direction

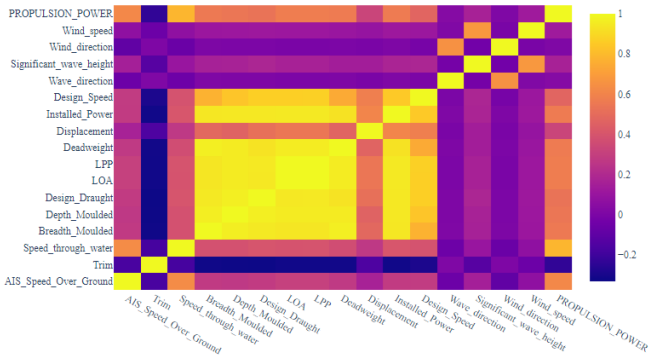

Figure 5. Correlation after data quality check

After the filter, the correlation between propulsion power and other features was improved significantly.The operational data for the 231 vessels are not unevenly distributed. For ships' data from noon reports, fewer records were recorded and reported less frequently. Therefore, the split of training and testing data considers the balancing of ships from different data sources. In general, the percentage of separation of training and test is around 80% and 20% based on number of records.

IV. RESULTS

A. Random Forest configuration

Compared to other machine learning methods, RF has fairly fewer number of hyperparameters to tune. In this study, 6 parameters were adapted to find the well performing model.

- max_features: The maximum number of features that RF is allowed to try in an individual tree. If 'auto', then max_features equals number of features. If 'sqrt', max_features equals the square root of number of features.
- max_depth: The maximum depth of an individual tree.
- n_estimators: The number of trees that will be built before taking the average of predictions or maximum voting. Higher number of trees can provide better prediction performance, but it will slow down the training.
- min_sample_leaf: The minimum number of samples in a leaf.
- min_samples_split: The minimum number of samples required to split an internal node.
- bootstrap: Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

In scikit-learn a large RF grid was created as the initial configuration. Then this configuration was adopted to part of the comprehensive data. The next step is searching the grid to find the best performing model. There are two types of searching functions in scikit-learn , one is 'GridSearchCV', the other is 'RandomizedSearchCV'. 'GridSearchCV' will go through all the intermediate combinations of hyperparameters. It makes the search computationally very expensive, but it can find the best combination based on the cross-validation score.. 'RandomizedSearchCV' solves the drawbacks of 'GridSearchCV', as it goes through only a fixed number of hyperparemetet settings. It tries to search randomly in the grid to find the best combination of hyperparameters. It helps to reduce unnecessary computation, but it does not guarantee to provide the best hyperparameter combination. Considering the computational difficulty, 'RandomizedSearchCV' was used in the initial RF grid. Based on the performance and the parameters of the best performing model, the scope of initial RF grid can be reduced. Fig. 5 shows the initial random forest grid.

```
{'max_features': ['auto', 'sqrt'],
'max_depth': [5, 10, 15, 20, 25, 30],
'n_estimators': [5, 10, 20, 50, 100, 200, 400, 600, 800, 1000]
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10],
'bootstrap': [True, False]}
```

Figure 5. Initial random forest grid

After previous step ,the optimized RF grid can be adapted in the Spark environment with MLlib. The naming and structure of hyperparameters in scikit-learn and Spark MLlib is slightly different. The match list between these two libraries can be found in table 2.

TABLE II. MATCH LIST OF LIBRARIES

Hyperparameters in scikit-learn	Hyperparameters in Spark Mllib
max_features	featureSubsetStrategy
max_depth	maxDepth
n_estimators	numTrees
min_samples_split	minInstancesPerNode
min_samples_leaf	
bootstrap	bootstrap

In Spark MLlib, 'min_sample_leaf' and 'min_samples_split' are merged and managed together through 'minInstancesPerNode'.Another point to be noted, MLlib only provide the normal grid search function which requires the grid to be searched should only be necessary. Otherwise, it can result in long search time and unnecessary

computational cost. Fig. 6 shows the optimized random forest grid.

```
{'numTrees': [10, 15, 20, 50, 100, 200],
 'minInstancesPerNode': [2, 4],
 'featureSubsetStrategy': ['sqrt'],
 'maxDepth': [20, 30],
 'bootstrap': [False]}
```

Figure 6. Optimized random forest grid

B. Best performing model

In this study, 3-fold cross-validation was adopted. R2 was used as evaluation metric in the training process. The RF model does not have overfitting problem[17]. During the grid search, when certain number of trees were created at certain stage, the performance of the model will stabilize around a specific value. From that moment, the test performance of random forest does not increase as the number of trees increases. In this study, the best performing model has the following parameters as shown in Fig. 7.

```
{'numTrees': 100
 'minInstancesPerNode': 2
 'featureSubsetStrategy': 'sqrt',
 'maxDepth': 30,
 'bootstrap': False}
```

Figure 7. Parameters of best berforming model

The operational propulsion power is a continuous value. Hence, it is taken as a regression problem. R2 score, also called the coefficient of determination was used as the major metric to evaluate the performance of the model. R2 score ranges from 0 to 1, and the higher the R2 score, the better and more accurate prediction was made. R2 score can be calculated from the following equations. SS_{tot} is the total sum of squares. SS_{res} is the regression sum of squares. f_i is the prediction result from the model.

$$SS_{tot} = \sum_i (y_i - \bar{y})^2 \quad (1)$$

$$SS_{res} = \sum_i (y_i - f_i)^2 \quad (2)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

The R2 of both training and test data were calculated. The R2 of training data is 0.9926. Because it is from training data, it is only used for reference. For the test data, the R2 is 0.9238 which is very good. Higher R2 values represent smaller differences between the target data and the predicted values. R2 also has limitations, as it does not represent the reliability of the model. Two kinds of visualization inspection were adapted to evaluate the performance of the model.

The timeseries plots show how the prediction follows the operation over time. In addition to the random forest model, the cubic law model based on load factor was added as reference[18]. Two container vessels with longer operation histories were selected. For both timeseries plots Fig. 8 and 10, parts of the time series were selected instead of all the historic records for better visibility. The scatter plots Fig. 9 and 11 can present the distribution of prediction against targets. The model which fits perfectly to the target will be a straight line along the diagonal.

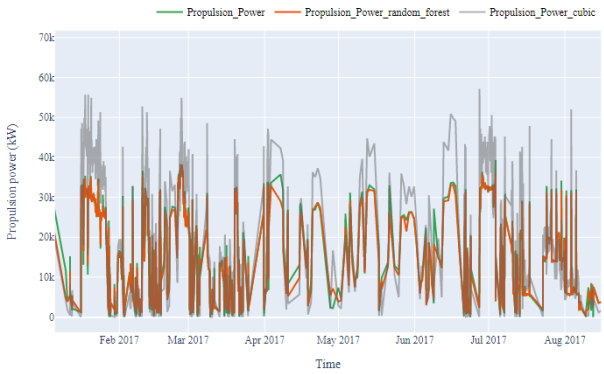

Figure 8. Ship 1 timeseries prediction

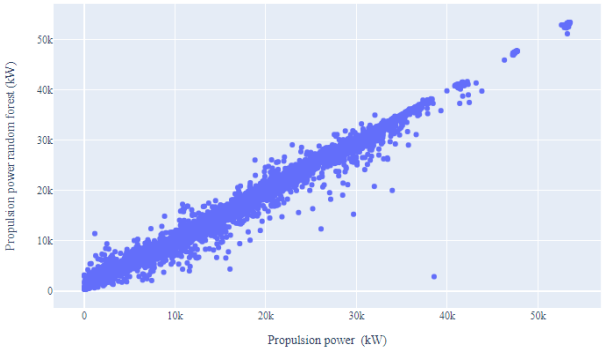

Figure 9. Ship 1 scatter distribution

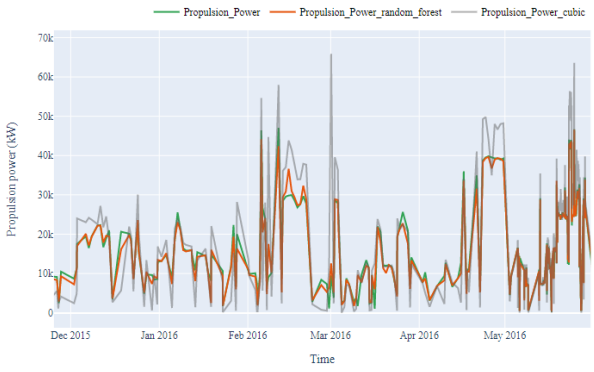

Figure 10. Ship 2 timeseries prediction

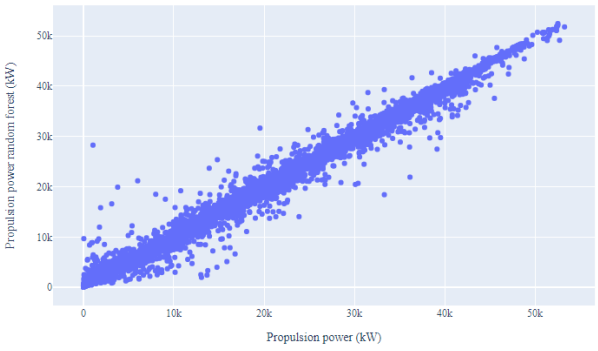

Figure 11. Ship 2 scatter distribution

The random forest model performs well on both the selected ships. Unlike the cubic law model, the random forest model takes more variables into account, such as ship characteristics and environment related variables. These

features can provide the model with more information about how the prediction should be made according to the specific condition. The cubic law model only takes the speed over ground as the dynamic input, however, in reality higher speed does not necessarily mean high propulsion power, and different environmental conditions can contribute significantly to the needed propulsion power.

Figure 12. Propulsion power and speed for ship 2

On the other hand, the operational propulsion power should follow the trend of operational speed. As shown in Fig. 12, the predicted propulsion power follows the operational speed well and align with the target propulsion power.

Figure 13. Test data with outliers scatter distribution

The performance of the RF model on the test data are visualized in Fig. 13. Several selected outliers were kept in the test data to evaluate the sensitivity of the model performance to the occurrence of outliers. These outliers have a normal operational speed, but the target propulsion power is extremely low which represent the deceleration process. On the left side of Fig. 13, some predictions with low target propulsion power, the random forest model predicts normal propulsion power based on the correct speed. They look like 'outliers', but the model is doing the right prediction with the normal operational speed.

C. *Life cycle management of the trained RF model*

As the digital transformation proceeds, huge amounts of data are being produced. More stakeholders are building cloud-based platforms or migrating their data to cloud-based infrastructure. Most of these data are stored in a way compatible with big data, e.g., Apache Spark, Google BigQuery and Apache Flink. Machine learning models should also be compatible with environments, can be trained, executed, and managed. The trained RF model in this study will be taken as an example to illustrate how the machine learning lifecycle with bigdata can be managed, as illustrated in Fig. 14.

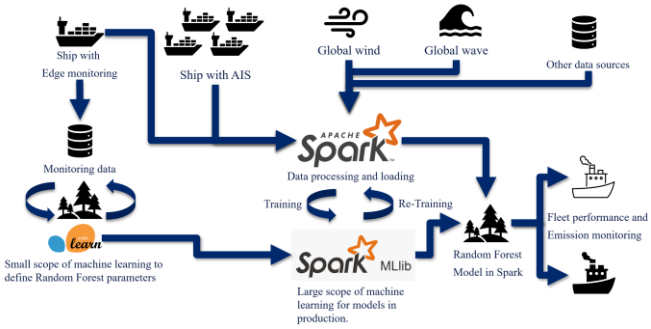

Figure 14. Machine learning life cycle framework with Spark

Not all ships are installed with edge monitoring devices. It means the amount of accurate monitoring data can be limited. These data can be used to search the boundary of the random forest grid. After that, the optimized grid can be adapted to the large scope of machine learning.

Different data sources can be saved in a data lake [19] which are then accessible by the Spark cluster. After that, complex data processing e.g., cleaning, interpolation and synchronization will be performed on the data. Then this comprehensive dataset will be used to train a Random Forest model with Spark MLlib. Because it was originally trained with Spark, it can be easily and efficiently applied with fleet monitoring data to do the monitoring of performance and emission. When a new data stream is arriving, the machine learning loop will be activated to re-train the model. If the new RF model performs better than the existing model, the new RF model will be deployed for further usage. Similarly, new edge monitoring data can also contribute to the selection of RF parameters.

V. Conclusion

In this paper, both single node and parallel machine learning methods were adopted to train a RF model for ship propulsion power prediction. Compared with many other machine learning methods, RF has many advantages, e.g., can handle missing values, is robust to outliers and requires no scaling. The performance of the best performing model was explored. From both timeseries and scatter visualization, the RF model can make accurate and reliable prediction. Several outliers were deliberately kept in the test data to check the reliability of the model. The RF model works well against these outliers. The result shows random forest is a feasible and robust method for ship propulsion power prediction on large datasets. The best performing model achieved a R2 score of 0.9238.

This paper also proposed an efficient machine learning framework for maritime big data based on Apache Spark environment. Both scikit-learn and Spark MLlib were used in the process to find the best configuration of hyperparameters. The trained RF model has high compatibility and can be executed on modern cloud platforms for fleet monitoring. With this framework, the search and training time of RF

model can be significantly reduced. This becomes more and more important when huge amount of data are continuously landing on cloud-based infrastructures.

REFERENCES

[1] IMO, "Marine Environment Protection Committee (MEPC) 77, 22-26 November 2021," 2021. .

[2] A. F. Molland, S. R. Turnock, and D. A. Hudson, "Ship Resistance and Propulsion," in *Ship Resistance and Propulsion*, 2011.

[3] Q. Liang, H. A. Tvete, and H. W. Brinks, "Prediction of vessel propulsion power from machine learning models based on synchronized AIS-, ship performance measurements and ECMWF weather data," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 929, no. 1, 2020, doi: 10.1088/1757-899X/929/1/012012.

[4] Q. Liang, H. A. Tvete, and H. W. Brinks, "Prediction of vessel propulsion power using machine learning on AIS data, ship performance measurements and weather data," *J. Phys. Conf. Ser.*, vol. 1357, no. 1, 2019, doi: 10.1088/1742-6596/1357/1/012038.

[5] Leo Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: https://doi.org/10.1023/A:1010933404324.

[6] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016, doi: 10.1007/s11749-016-0481-7.

[7] O. Bodunov, F. Schmidt, A. Martin, A. Brito, and C. Fetzer, "Grand challenge: Real-time destination and ETA prediction for maritime trac," *DEBS 2018 - Proc. 12th ACM Int. Conf. Distrib. Event-Based Syst.*, pp. 198–201, 2018, doi: 10.1145/3210284.3220502.

[8] H. Zhong, X. Song, and L. Yang, "Vessel Classification from Space-based AIS Data Using Random Forest," in *Proceedings - 2019 5th International Conference on Big Data and Information Analytics, BigDIA 2019*, 2019, pp. 9–12, doi: 10.1109/BigDIA.2019.8802792.

[9] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines," *Ore Geol. Rev.*, vol. 71, pp. 804–818, 2015, doi: 10.1016/j.oregeorev.2015.01.001.

[10] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer, "Robust and Accurate Shape Model Fitting Using Random Forest Regression Voting BT - Computer Vision – ECCV 2012," 2012, pp. 278–291.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, pp. 12–2825--2830, 2012, Accessed: May 29, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html.

[12] "Apache Spark's Machine Learning Library (MLlib)." https://databricks.com/glossary/what-is-machine-learning-library.

[13] X. Meng *et al.*, "MLlib: Machine learning in Apache Spark," *J. Mach. Learn. Res.*, vol. 17, pp. 1–7, 2016.

[14] K. Gunnar Aarsæther and T. Moan, "Estimating navigation patterns from AIS," *J. Navig.*, vol. 62, no. 4, pp. 587–607, 2009, doi: 10.1017/S0373463309990129.

[15] USNA, "Resistance and powering of ships," *Resist. powering ships*, pp. 1–46, 2002.

[16] A. Mjelde, K. Martinsen, M. Eide, and O. Endresen, "Environmental accounting for Arctic shipping - A framework building on ship tracking data from satellites," *Mar. Pollut. Bull.*, vol. 87, no. 1, pp. 22–28, Oct. 2014, doi: 10.1016/j.marpolbul.2014.07.013.

[17] M. R. Segal, "Machine Learning Benchmarks and Random Forest Regression Publication Date Machine Learning Benchmarks and Random Forest Regression," *Cent. Bioinforma. Mol. Biostat.*, p. 15, 2004, [Online]. Available: https://escholarship.org/uc/item/35x3v9t4.

[18] R. Adland, P. Cariou, and F. C. Wolff, "Optimal ship speed and the cubic law revisited: Empirical evidence from an oil tanker fleet," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 140, no. October 2019, p. 101972, 2020, doi: 10.1016/j.tre.2020.101972.

[19] AWS, "What is a data lake?," *Documentation of Amazon Web Services*. https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/.