

Article

# Towards the Evaluation of Date Time Features in a Ship Route Prediction Model

Angelica Lo Duca \* and Andrea Marchetti

Institute of Informatics and Telematics, National Research Council, via G. Moruzzi 1, 56124 Pisa, Italy

\* Correspondence: angelica.loducal@iit.cnr.it; Tel.: +39-050-3158292

**Abstract:** Ship Route Prediction (SRP) is an algorithm that allows assessing the future position of a ship using historical data, extracted from AIS messages. In an SRP task, it is very important to select the set of input features, used to train the model. In this paper, we try to evaluate if time-dependent features are relevant in an SRP model, based on a K-Nearest Neighbor classifier, through a practical experiment. In practice, we build two models, with and without the Date Time features, and for both models, we calculate some performance metrics and the SHAP value. Tests show that although the model with the Date Time features outperforms the other model in terms of evaluation metrics, it does not in the practical experiments.

**Keywords:** machine learning; Ship Route Prediction; feature engineering; marine data science



**Citation:** Lo Duca, A.; Marchetti, A. Towards the Evaluation of Date Time Features in a Ship Route Prediction Model. *J. Mar. Sci. Eng.* **2022**, *10*, 1130. <https://doi.org/10.3390/jmse10081130>

Academic Editor: Mihalis Golias

Received: 29 July 2022

Accepted: 15 August 2022

Published: 17 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A Ship Route Prediction (SRP) model is a vastly complex system that incorporates the natural variability of a system coupled with the determination of output by utilizing the best input features. The model offers the possibility to predict the movement of ships within a specific area determined by the boundaries of a sea chart. SRP is utilized in commercial shipping, fishing, and naval activities, as well as in military planning. As pointed out by Montiel et al. in [1], SRP can be used in different scenarios, such as enhancing the safety of a voyage [2], simplifying route planning, avoiding collisions [3,4], and reducing costs of the voyage [5]. SRP can be used to give information on accepted lanes and sortie areas and generally provides information crucial to ocean traffic management. Such a module gives details when a vessel is scheduled for arrival at a certain port and its next departure location after completing loading/unloading operations at said port.

To perform SRP, first of all, some statistical data on the movement of the ship should be collected. Such data can be extracted from AIS messages [6] and includes at least the current status of the ship, which contains the ship's coordinates, speed, heading, and size. Other data could be added to the current status of the ship, such as the weather conditions and the current date.

A vast literature exists about SRP [1,7,8]. In this paper, only the most representative papers are described. Algorithms for SRP can be classified into three categories: (1) points-based, which predicts only the immediately next position of the ship; (2) trajectory-based, which predicts the whole trajectory; and (3) hybrid-based, which combines the previous two categories.

Points-based algorithms split the area to be monitored in different non-overlapping cells, having all the same size. Given the current status of a ship, they calculate the probability that each cell of the area will be occupied after a given period of time. This analysis is carried out on the basis of historical data, generally extracted from AIS messages. This class of algorithms is implemented through different approaches such as neural networks [9–11], associative rules [12], Kernel density estimation [13], and K-NN [14].

Trajectory-based algorithms exploit historical data to build clusters, to extract and classify routes. Routes are then represented through a model, which can be improved

through the inclusion of waypoints (harbors, offshore platforms, and entry and exit points in the area). Examples of trajectory-based algorithms exploit extended Kalman filter [15], similarity-based approach and kernel-based machine learning methods [16], synthetic route knowledge [17], a data-driven nonparametric Bayesian model based on a Gaussian Process [18], neural networks [19], deep learning [20], and generative models [21].

While building an SRP algorithm, it is very important to choose the input features to provide as input to the model. All the previous literature does not consider the Date Time as a candidate input feature. This paper evaluates whether the use of one or more Date Time input features could produce a robust model in practical experiments or not. Two models are built, both trained and evaluated on the same training and test sets: the first considers the Date Time as an input feature, and the second does not. Some evaluation metrics are calculated for both algorithms, including precision, recall, and accuracy and then are compared. In addition, a practical experiment is run, with real data, and both algorithms for predictions are used. The best algorithm is finally shown. The SHAP value [22] is also calculated for each model, to understand the contribution of each feature to the final output.

This work is an improvement of the previous work [23], where a comparison among different multiclass classification algorithms was performed to solve the SRP problem. The SRP algorithm implemented in the previous work was carried out within the Optical/SAR data and System Integration for Rush Identification of Ship models (OSIRIS) project (The OSIRIS project: <http://si.isti.cnr.it/index.php/hid-project-category-list/44-project-osiris-page> (accessed on 10 August 2022)), which aimed at building a surveillance maritime system. With respect to the previous work, there is a focus on the K-Nearest Neighbors classifier, which is used to evaluate the importance of the Date Time feature. Retraining of the algorithm with a bigger dataset is also performed as well as a description of a practical experiment.

The paper is organized as follows: Section 2 describes briefly the OSIRIS-FO project, which is the follow on of the OSIRIS project, while Section 3 formulates the problem, by also describing how the training and test sets are built. Section 4 discusses the results. Finally, Section 5 gives conclusions and possible improvements for the future.

## 2. The OSIRIS-FO Project

The work presented in this paper was implemented within the OSIRIS Follow-On (OSIRIS-FO) project (The OSIRIS-FO project: <http://si.isti.cnr.it/index.php/hid-project-category-list/206-project-osirisfo> (accessed on 10 August 2022)), funded by the European Space Agency. OSIRIS-FO is the enhancement of the previous project, called OSIRIS (The OSIRIS project: <http://si.isti.cnr.it/index.php/hid-project-category-list/44-project-osiris-page> (accessed on 10 August 2022)). The main objective of OSIRIS-FO involves the implementation of a system to support decision-making processes for maritime domain awareness.

Figure 1 shows the OSIRIS architecture. OSIRIS focuses on a given area of interest (AoI) and elaborates on satellite images within that AoI, to extract ships from such images (Ship Detection). Once detected, ships are also classified (Ship Classification), according to some criteria (e.g., ship length, width, class of ship, etc.) Then, for each ship, the current course over the ground and speed are calculated (Ship Kinematic Estimation). The extracted information and the AIS messages are given as input Ship Route Prediction, which calculates the next position of the ship after one hour. Finally, Data Fusion recognizes exactly every ship name within the image, by combining data contained in the AIS database with those extracted by Ship Detection [24].

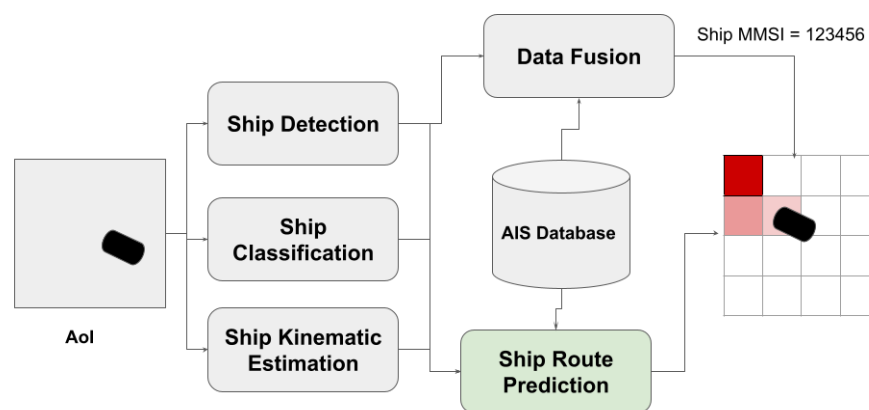


Figure 1. The OSIRIS-FO architecture.

The SRP problem can be solved through the division into cells of the AoI and the subsequent identification of which cell will be occupied by the ship after a certain time interval, given the current status of a ship. This is a multiclass classification problem, where every target class corresponds to a cell of the grid, identified by a row and a column. As input features, the current status of the ship, which includes the current latitude and longitude, the current speed over the grounds, the current course over the ground h, the class of a ship, and, optionally, the date are considered. In this paper, only the differences with respect to our previous work [23] are described, and thus, for further details, you can refer to it.

### 3. Methodology

The objective of this study is to test whether adding a Date Time feature as an additional input feature improves the performance of the SRP algorithm developed in [23]. In particular, the previous model was based on a K-Nearest Neighbors algorithm, which received as input the following features relating to the current state of the ship: position, speed, direction, and class of the ship (small or big). In order to verify if the addition of a Date Time feature improved the performance of the algorithm, two versions of the KNN were implemented. The first version is trained by considering the date as an input feature (KNN With Date or shortly KNN-WD), and the second version does not (KNN). The two versions were trained on the same training set, with the only difference that in the case of the KNN-WD the information relating to the date was also considered. The performance of the two algorithms is calculated in two ways. The first way uses classic performance measurement metrics (precision, recall, and accuracy). The second method is based on a more heuristic approach, which implements a practical experiment.

The described approach includes the following steps: (a) data preparation, (b) model training and evaluation, (c) a practical experiment. This methodology follows the standard approach based on splitting the dataset into two parts, training and test sets, and then using a set of unseen samples for validation. In this paper, data extracted from the practical experiment are used as validation set. An alternative to this approach could be splitting the original dataset in three parts, training, test, and validation sets. However, this approach could propagate a possible bias present in the original dataset.

Before describing the followed methodology in detail, here a short glossary of used terms used in this paper is given:

- AIS messages—the messages periodically by ships.
- AoI—the Area of Interest. In this paper, the area located under the Malta isle is considered.
- KNN—the K-Nearest Neighbors algorithm. In this paper this acronym is used to indicate the algorithm which receives as input the following features: geographical coordinates, speed, course, and ship class.

- KNN-WD—the K-Nearest Neighbors algorithm with Date Time features. In this paper, this acronym is used to indicate the algorithm which receives as input the following features: geographical coordinates, speed, course, ship class, and date time.

### 3.1. Data Preparation

The dataset used to train the two algorithms is built from 188,130,529 AIS messages extracted from Astra Paging, covering a time interval from 26 December 2015 to 24 December 2017 and referring to the AoI near the isle of Malta. An AoI with a cell size of  $0.1 \times 0.1$  degrees was considered.

To prepare the dataset for model training and evaluation, a new dataset is created, which is a subset of the original dataset, which is cleaned and balanced. The following pre-processing techniques have been applied to the raw dataset: (a) cleaning, (b) discretization, (c) normalization, and (d) balancing.

#### 3.1.1. Data Cleaning

Data cleaning involves the deletion of all records which do not satisfy some criteria. In this case, the following rows were deleted:

- Speed less than 0.1 knots.
- Speed greater than 60 knots.
- Course greater than 360 degrees.
- Records where the MMSI is present only once.

Analysis could be carried out to understand why in some cases speed is greater than 60 knots, and the course is greater than 360 degrees. At the moment, these records have been removed. However, they could be replaced by adopting some of the most common techniques used to deal with missing values, such as the replacement with the average value. This task is demanded as future work on this analysis if it is needed. At the end of the cleaning process, 76,039,396 records remained, which corresponded to 2397 classes, each representing a different cell of the AoI. The number of classes could be reduced to simplify the problem, for example, grouping adjacent cells. However, this grouping strategy could produce a coarser grain resolution of the AoI, thus making quite difficult to detect the exact position of the ship in the ocean. The more cells, the more accurate the predicted position of the ships.

#### 3.1.2. Discretization

Discretization converts values belonging to a continuous domain to quantized values. Through discretization, the number of possible values assumed by a feature is reduced, thus the estimation algorithm can perform better. With respect to the previous work [23], discretization and transformation of the input features was carried out as follows.

Discretization was carried out by dividing the course into 8 clock faces, as illustrated by Table 1 [15]:

**Table 1.** Course discretization into 8 clock faces.

Course over the Ground (Degrees)	Discrete Value	Description
337.5–22.5	0	Nord
22.5–67.5	1	Nord Est
67.5–112.5	2	Est
112.5–157.5	3	Sud Est
157.5–202.5	4	Sud
202.5–247.5	5	Sud West
247.5–292.5	6	West
292.5–337.5	7	North West

All the possible speeds were grouped into 4 slots (Table 2): slow, medium, high, and very high [15].

**Table 2.** Speed discretization into four slots.

Speed Interval (Knots)	Discrete Value	Description
[0.5–3]	0	slow
(3, 14]	1	medium
(14, 23]	2	high
(23, 60]	3	very high

Finally, latitude and longitude were also discretized, by setting the latitude to the value of the row in the matrix, representing the AoI, and the longitude to the column.

### 3.1.3. Normalization

Data Normalization involves adjusting values measured on different scales to a common scale. Different techniques exist to perform normalization. In this work, single feature scaling was applied. Single Feature Scaling converts every value of a column into a number between 0 and 1. The new value is calculated as the current value divided by the max value of the column:

$$x_{new} = \frac{x_{current}}{x_{max}}$$

### 3.1.4. Data Balancing

Since the dataset was imbalanced, the number of samples for each target class was set to 1000. The following balancing strategy was adopted: (1) undersample the classes with more than 1 M records through Random Undersampling; (2) undersample the classes with a number of records between 1000 and 1 M through Cluster Centroids; (3) oversample the classes with less than 1000 records through Random Oversampling. The final balanced dataset contained 2,397,000 records. Table 3 resumes the balancing strategy.

**Table 3.** Balancing strategy.

Number of Classes	Number of Records	Balancing Strategy
2	>1 Million	Random Undersampling
1570	>1000, <1 Million	Cluster Centroids
825	<1000	Random Oversampling

## 3.2. Model Training and Evaluation

The two Machine Learning models were tuned through Grid Search with Cross-Validation with Kfold = 5. Table 4 shows the tuned parameters and the best value for each parameter.

**Table 4.** Tuned parameters for KNN and KNN-WD.

Parameter	Ranges	Best Value
n_neighbors	3, 4	3
weights	uniform, distance	distance

The total number of records is 2,397,000. The training set contains 2,157,300 records and the test set the remaining 239,700 records.

Table 5 shows the performance of both the algorithms.

**Table 5.** Precision, recall, and accuracy of the two algorithms.

Metric	KNN-WD (with Date)	KNN	Percentage Increase
Precision	0.84	0.64	23.8%
Recall	0.85	0.63	24.8%
Accuracy	0.85	0.63	24.8%

The table shows that in terms of evaluation metrics, KNN-WD outperforms KNN with a percentage increase of more than 23% in all the cases. This means, that theoretically, the use of the Date Time feature should improve the performance of an SRP algorithm.

### 3.3. A Practical Experiment

To test the performance of the algorithms in a real case, a practical experiment was set up. The goal of the experiment was to demonstrate how both algorithms work in a practical scenario. The practical experiment used a satellite image, taken in a period covered by the AIS data with which the algorithm was trained. The image was downloaded from the Copernicus Open Access Hub (<https://scihub.copernicus.eu/>, accessed on 16 August 2022) (The extracted image is the S1A\_IW\_GRDH\_1SDV\_20170525T165531\_20170525T165556\_016741\_01BCE3\_614D.SAFE, relating to an area south of Sicily and referring to the date 25 May 2017 at 16:55:56). Through the various modules of the OSIRIS project, the current status of the ship was extracted from the image, including position, speed, direction, and class of the ship. The Ship Detection module detected about 100 ships in the image. Then both the KNN and KNN-WD algorithms were run to calculate the prediction after one hour and a visual procedure was performed to establish whether the predictions were corrected.

Through the Data Fusion module, also developed within the OSIRIS project, a mapping between the AIS data and the ships extracted from the image was carried out. This permitted to reconstruct the correct path of the ship and verify the correctness of the predicted values by SRP. Among the 100 ships identified by the Ship Detection module, the Data Fusion module recognized the MMSI of only 71 ships. Among them, only 32 ships had a speed greater than 0.5 knots. Among them, only 26 ships had a sufficient number of AIS messages in the AIS dataset, which permitted us to build the ship trajectory and thus predict the ship position after one hour.

For each ship, the actual position of a ship after one hour was extracted from the AIS dataset and then compared with the predicted position by the two algorithms. Results were classified as follows:

- Correct prediction—the predicted cell contained exactly the actual position of the ship after one hour.
- Acceptable prediction—the predicted cell was adjacent to the actual position of the ship after one hour.
- Wrong prediction—all the other cases.

Table 6 shows the results of the tests for both the algorithms.

**Table 6.** Results of the experiment.

Type of Result	KNN-WD (Number of Ships)	KNN (Number of Ships)
Correct	1	13
Acceptable	13	13
Wrong	12	0

Results shown in Table 6 are in contrast with those shown in Table 5. A possible explanation of this divergence will be discussed in the next section.

The remainder of this section shows the results for some ships. For each ship, the predicted values are represented through squares, and the ship trajectory is shown as dots





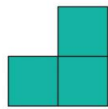
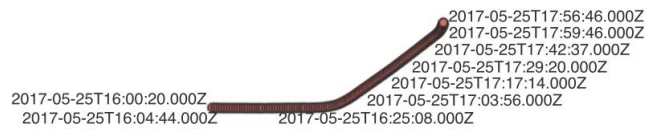


Figure 5. The output of SRP for ship 81, when using KNN-WD.

#### 4. Discussion

Model evaluation showed that KNN-WD outperformed KNN, while the practical experiment demonstrated the contrary.

The previous experiments show that there is a discrepancy between the calculated metrics and the performance of the algorithms in real use-cases. In fact, in the real use-cases, KNN outperforms KNN-WD, although in terms of performance metrics, KNN-WD outperforms KNN. Probably this is due to a different distribution between the data of the training/test set and those of the real experiments.

To better understand the phenomenon, two types of analysis were performed: (a) distribution of the date features, (b) SHapley Additive exPlanations (SHAP) value. The two types of analysis are discussed separately.

##### 4.1. Distribution of the Date Features

An analysis of the distribution of the features related to the date in the training set was performed, as well as where the date of the experiments is placed. Figure 6 shows the day sin, day cosine, hour sin, and hour cosine distribution, and, in red, where the date of the real experiments falls.

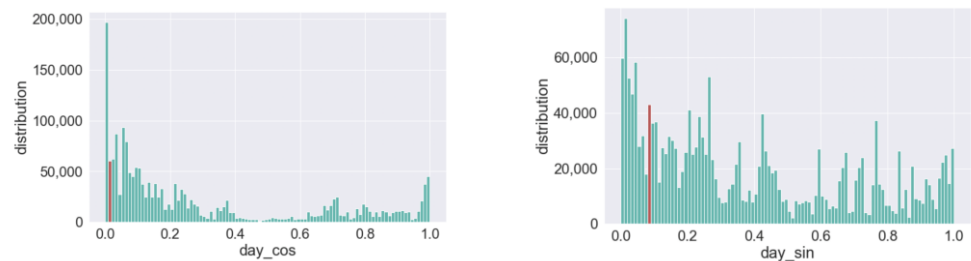
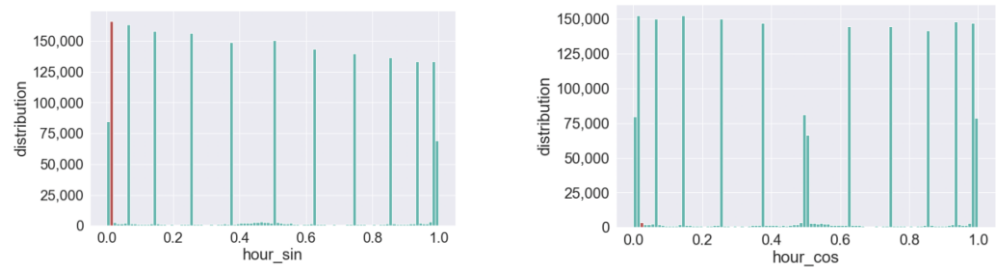


Figure 6. Cont.





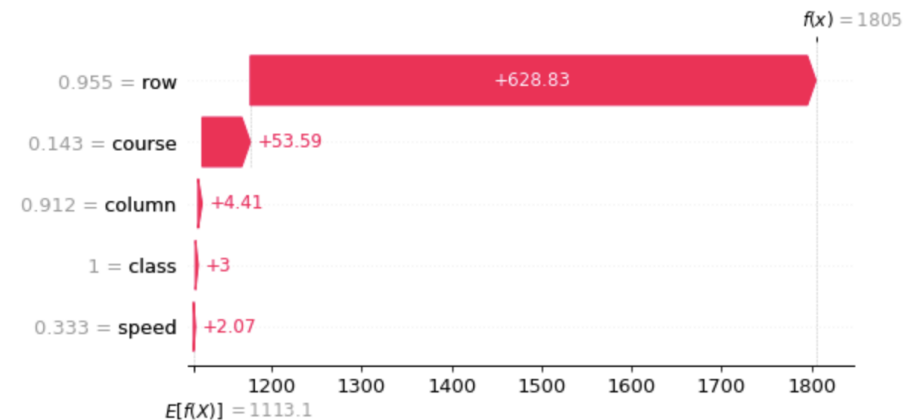
**Figure 6.** The distribution of the date features (day sin, day cosine, hour sin, and hour cosine).

The hour cosine associated with the real experiment falls into an interval where there are few samples. This probably means that there are not enough samples in the training test to correctly use the date features in the real experiments. This would justify why KNN outperforms KNN-WD. In other words, the training/test set and the real experiments seem to not follow the same distribution, thus producing overfitting of the training/test set. However, this is not completely true, because the satellite image was extracted from dates available in the AIS dataset.

To solve the previous problem, an alternative could be to search for a dataset where there are enough samples for all the Date Time features. However, it could not be easy to search for a similar dataset because a balance in the Date Time features may not correspond to an equal balance in the number of samples per target class. Too many factors should be considered, which require a separate study. This could be investigated as a future work.

#### 4.2. SHAP Value

SHAP is a method extracted from game theory to calculate the contribution of each input feature to predict an output [23]. Figures 7 and 8 show the contribution of each input feature to produce the value predicted by KNN and KNN-WD for ship 26 of the experiment described in the previous section.



**Figure 7.** The contribution of each input feature to build the output predicted by KNN for ship 26.

In the previous example, KNN predicts 1805 (correct) as the target class, while KNN-WD 1124 (acceptable). Each model starts from a standard value, identified as  $E[f(x)]$ , and then it adds or subtracts the contribution of each feature. In KNN, the row is the class giving the greatest contribution, while in the KNN-WD the Date Time features contribute to lowering the value of the predicted class. The presence of many negative contributions in KNN-WD justifies why KNN-WD tends to predict a very low value, which is not the case in KNN, which is very close to the expected output.

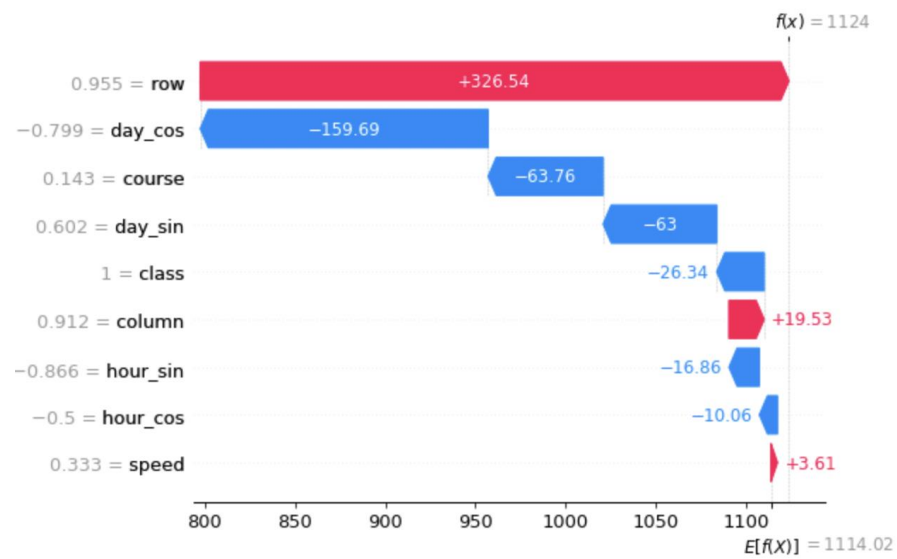


Figure 8. The contribution of each input feature to build the output predicted by KNN-WD for ship 26.

Figures 9 and 10 are similar to Figures 7 and 8, but they show results for ship 81.

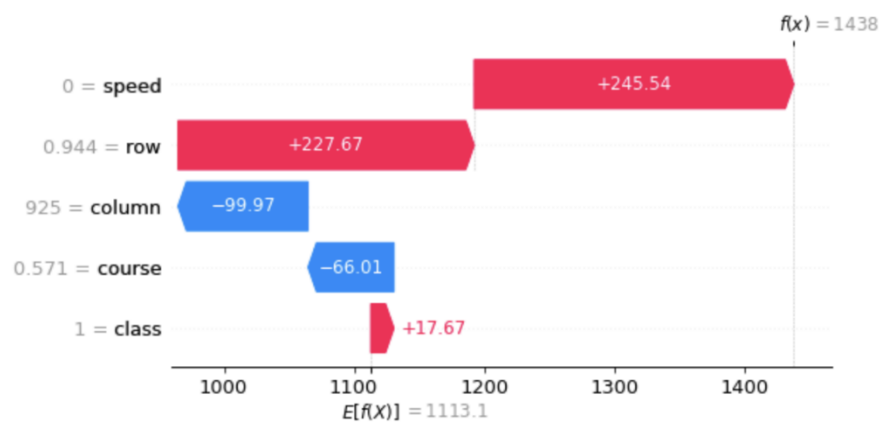


Figure 9. The contribution of each input feature to build the output predicted by KNN for ship 81.

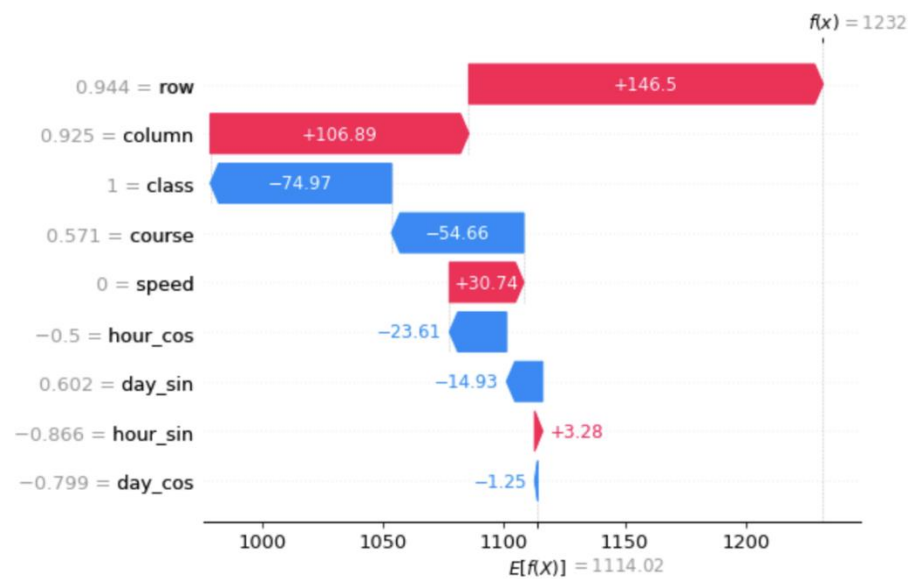


Figure 10. The contribution of each input feature to build the output predicted by KNN-WD for ship 81.

In Figure 9 the predicted value is 1438 (acceptable), which is built by the sum of positive features (speed, row, and class) and negative ones (column and course). In Figure 10, the predicted value is 1232 (wrong). Again, the Date Time features contribute to lower the predicted value, with the exception of hour\_sin, which gives a positive contribution.

Both previous examples show that Date Time features give a negative contribution to the final predicted output.

Figures 11 and 12 show the average SHAP for all the 26 ships considered in the experiment, for KNN and KNN-WD respectively. In both cases, the row is the most important input feature. In KNN-WD, the day\_sin feature is the second most important feature, thus giving a strong contribution to the final prediction. Then, in both cases, course, column, and speed make a contribution in descending order of importance. In both cases, in the last place, there is the class.

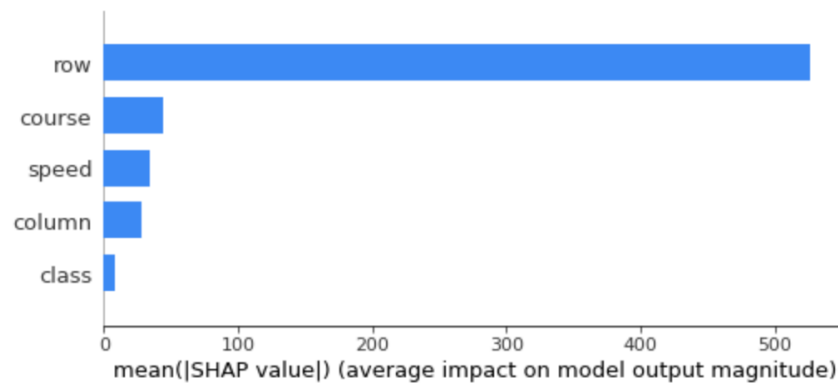


Figure 11. The average SHAP value for each input feature of the experiment for KNN.

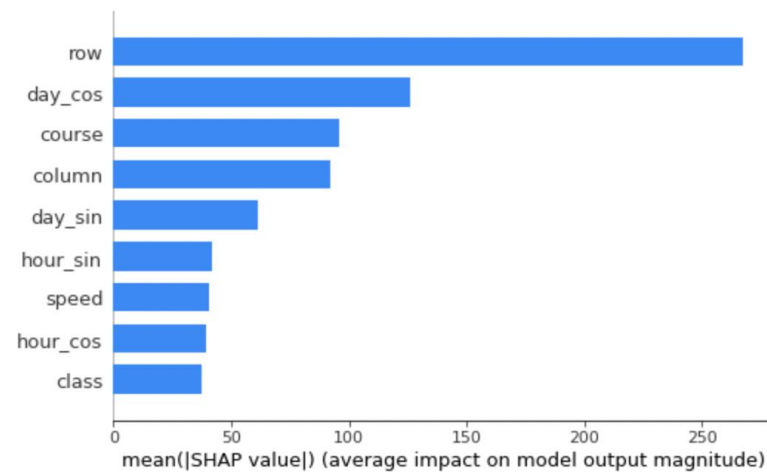


Figure 12. The average SHAP value for each input feature of the experiment for KNN-WD.

The previous analysis showed that using Date Time as input features strongly affects the behavior of an algorithm, and thus, great attention should be paid while using a feature of this type.

### 5. Conclusions and Future Work

This paper discussed the importance of time-dependent input features for a classification problem. A practical experiment was implemented, showing that although KNN-WD outperforms KNN in terms of performance metrics, KNN behaves better than KNN-WD in practical experiments. This is probably due to the fine-grained date feature: in the training/test set, there were many samples for the considered period of time, while in the experiments, there were few samples for the considered period of time.

The most important findings of this paper include:

- Special attention should be paid when dealing with Date Time features. In fact, the temporal conditions present in the training set may no longer be true at run-time.
- The calculation of the SHAP value in the experiments showed that the Date Time features greatly influence the determination of the results, giving almost always a negative contribution, which tends to lower the final predicted value.

The described results rely onto an empirical analysis, which focuses on a practical experiment. As a future work, a more theoretical approach could be defined, to demonstrate the influence of Date Time features on the behavior of a Machine Learning algorithm.

To improve the KNN-WD algorithm, a coarse grain discretization of dates could be performed, e.g., use season instead of day of year, split 24 h in 4 slots and so on. This could lead to more samples in each time slot and thus reduce the impact of the date in the SHAP value calculation.

Finally, as pointed out in Section 4.1, a deeper analysis could be carried out to build a training set with a more balanced distribution of the Date Time input features. This should also take into account a balance in the target classes.

**Author Contributions:** Conceptualization, A.L.D. and A.M.; methodology, A.L.D. and A.M.; software, A.L.D.; validation, A.L.D.; formal analysis, A.L.D. and A.M.; investigation, A.L.D. and A.M.; resources, A.L.D. and A.M.; data curation, A.L.D. and A.M.; writing—original draft preparation, A.L.D.; writing—review and editing, A.L.D.; visualization, A.L.D.; supervision, A.M.; project administration, A.M.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the OSIRIS-FO project, which is the follow-on of ESA's GSTP Project OSIRIS (Optical/SAR data and system Integration for Rush Identification of Ship models, 2016–2018).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Montiel, E.A.R.; Klunk, R.F.; Tijan, E.; Jović, M. Using Automatic Identification System Data in Vessel Route Prediction and Seaport Operations. *J. Marit. Transp. Sci.* **2021**, *61*, 45–56. [[CrossRef](#)]
2. Murray, B.; Perera, L.P. A data-driven approach to vessel trajectory prediction for safe autonomous ship operations. In Proceedings of the 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 24–26 September 2018; pp. 240–247. [[CrossRef](#)]
3. Huang, Y.; Chen, L.; Chen, P.; Negenborn, R.R.; van Gelder, P.H.A.J.M. Ship collision avoidance methods: State-of-the-art. *Saf. Sci.* **2020**, *121*, 451–473. [[CrossRef](#)]
4. Zaccone, R.; Martelli, M. A collision avoidance algorithm for ship guidance applications. *J. Mar. Eng. Technol.* **2020**, *19*, 62–75. [[CrossRef](#)]
5. International Maritime Organization. International Convention for the Safety of Life at Sea (SOLAS), Chapter V: Safety of Navigation, Regulation, Volume 19. 2002. Available online: [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/343175/solas\\_v\\_on\\_safety\\_of\\_navigation.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/343175/solas_v_on_safety_of_navigation.pdf) (accessed on 16 August 2022).
6. Moreira, L.; Vettor, R.; Soares, C.G. Neural Network Approach for Predicting Ship Speed and Fuel Consumption. *J. Mar. Sci. Eng.* **2021**, *9*, 119. [[CrossRef](#)]
7. Tu, E.; Zhang, G.; Rachmawati, L.; Rajabally, E.; Huang, G.-B. Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1559–1582. [[CrossRef](#)]
8. Cazzanti, L.; Pallotta, G. Mining Maritime vessel traffic: Promises, challenges, techniques. In Proceedings of the OCEANS 2015, Genova, Italy, 18–21 May 2015; pp. 1–6.
9. Zhang, S.; Wang, L.; Zhu, M.; Chen, S.; Zhang, H.; Zeng, Z. A Bi-directional LSTM Ship Trajectory Prediction Method based on Attention Mechanism. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; Volume 5, pp. 1987–1993. [[CrossRef](#)]
10. Park, J.; Jeong, J.; Park, Y. Ship Trajectory Prediction Based on Bi-LSTM Using Spectral-Clustered AIS Data. *J. Mar. Sci. Eng.* **2021**, *9*, 1037. [[CrossRef](#)]

11. Rhodes, B.J.; Bomberger, N.A.; Zandipour, M. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness. In Proceedings of the 2007 10th International Conference on Information Fusion, Quebec, QC, Canada, 9–12 July 2007; pp. 1–8. [[CrossRef](#)]
12. Deng, F.; Guo, S.; Deng, Y.; Chu, H.; Zhu, Q.; Sun, F. Vessel track information mining using AIS data. In Proceedings of the International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), Beijing, China, 28–29 September 2014; p. 16.
13. Ristic, B. Detecting Anomalies from a Multitarget Tracking Output. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 798–803. [[CrossRef](#)]
14. Virjonen, P.; Nevalainen, P.; Pahikkala, T.; Heikkonen, J. Ship movement prediction using kNN method. In Proceedings of the Baltic Geodetic Congress (BGC Geomatics), Iosztyn, Poland, 21–23 June 2018; pp. 304–309. [[CrossRef](#)]
15. Perera, L.P.; Oliveira, P.; Guedes Soares, C. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1188–1200. [[CrossRef](#)]
16. de Vries, G.K.D.; van Someren, M. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Syst. Appl.* **2012**, *39*, 13426–13439. [[CrossRef](#)]
17. Volkova, T.; Balykina, Y.; Bespalov, A. Predicting Ship Trajectory Based on Neural Networks Using AIS Data. *J. Mar. Sci. Eng.* **2021**, *9*, 254. [[CrossRef](#)]
18. Tafa, L.N.; Su, X.; Hong, J.; Choi, C. Automatic Maritime traffic synthetic route: A framework for route prediction. In *International Symposium on Pervasive Systems, Algorithms and Networks*; Springer: Cham, Germany, 2019; pp. 3–14. [[CrossRef](#)]
19. Murray, B.; Perera, L.P. A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data. *Ocean Eng.* **2020**, *209*, 107478. [[CrossRef](#)]
20. Rong, H.; Teixeira, A.; Soares, C.G. Ship trajectory uncertainty prediction based on a Gaussian Process model. *Ocean Eng.* **2019**, *182*, 499–511. [[CrossRef](#)]
21. Sørensen, K.A.; Heiselberg, P.; Heiselberg, H. Probabilistic Maritime Trajectory Prediction in Complex Scenarios Using Deep Learning. *Sensors* **2022**, *22*, 2058. [[CrossRef](#)] [[PubMed](#)]
22. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
23. Lo Duca, A.; Marchetti, A. Exploiting multiclass classification algorithms for the prediction of ship routes: A study in the area of Malta. *J. Syst. Inf. Technol.* **2020**, *22*, 289–307. [[CrossRef](#)]
24. Reggiannini, M.; Righi, M.; Tampucci, M.; Lo Duca, A.; Bacciu, C.; Bedini, L.; D’Errico, A.; Di Paola, C.; Marchetti, A.; Martinelli, M.; et al. Remote Sensing for Maritime Prompt Monitoring. *J. Mar. Sci. Eng.* **2019**, *7*, 202. [[CrossRef](#)]