Application of decision trees to predict damage consequences during the progressive flooding

L. Braidotti
Faculty of Engineering, University of Rijeka, Rijeka, Croatia
University of Trieste, Trieste, Italy

J. Prpić-Oršić
Faculty of Engineering, University of Rijeka, Rijeka, Croatia

M. Valčić
Faculty of Engineering, University of Rijeka, Rijeka, Croatia
University of Zadar, Zadar, Croatia

ABSTRACT: In the last decade, the application of progressive flooding simulations improved the decision support available onboard in case of flooding. However, all these Decision Support Systems (DSS) rely on flooding sensors, thus cannot be adopted on the large majority of the existing fleet without a costly retrofit of the flooding detection system. Here an alternative has been proposed to assess the main flooding consequences from the time evolution of the floating position, which can be recorded with a very basic set of sensors. Here the decision trees are employed to assess the final fate of the ship, the damaged compartments and estimate the time-to-flood. Decision trees are here trained by means of two types of databases of progressive flooding simulations: one based on Monte Carlo (MC) generation of damages according to SOLAS probability distributions and a parametric one. The method has been tested on barge geometry employing another MC database for validation purposes.

1 INTRODUCTION

In recent years, the importance of decision support during the progressive flooding of a damaged ship has been highlighted by many severe accidents. In this contest, one of the most vulnerable ship types is represented by large passenger vessels (Braidotti et al. 2021a), whose complex non-watertight subdivision, limited stability reserve and limited freeboard at bulkhead deck lead to a difficult prediction of the flooding consequences. Hence, after damage due to collision or grounding it is very handy for the master to have his decisions supported by evidence since thousands of human lives can be a stake.

The disaster of Costa Concordia clearly showed the inadequacy of information to the master required by late international rules. At that time, it was not easy even identifying the breached compartments. Moreover, the consultation of mandatory onboard documentation regarding damage stability/control can further waste time. Some decision support could be provided by the loading computer system (Ruponen et al. 2019) or by Decision Support Systems (DSS) for damage control (Hu & Ma 2008, Kang et al. 2017). However, systems based on lost-buoyancy calculation are not capable to assess the intermediate stages of flooding, which might lead to large heeling angles or even ship capsize before reaching the final equilibrium position.

The introduction of a mandatory flooding detection system on passenger vessels that come into force on July 1st 2010 (IMO 2008) improved a fast achievement of situational awareness for new buildings. Furthermore, it opened new possibilities for the direct application of quasi-static progressive flooding simulation codes onboard (Ruponen et al. 2012, Dankowski & Krüger 2012, Rodrigues, & Guedes Soares, 2015). The flooding sensors fitted in each ship internal space permit the assessment of the damage dimension and location (Ruponen et al. 2017), which serves as input for the simulation of the progressive flooding process in the time-domain. In this manner, the damage consequences can be reliably forecasted to support master decisions immediately after damage occurrence, reducing the reaction time for damage control or ship abandonment, if required.

However, up to now, the large majority of the world fleet is not equipped with flooding sensors, being still vulnerable due to deficient emergency decision support. Besides, the high cost of flooding detection system retrofit hinders the application of time-domain tools on older vessels. To make available the essential information during a flooding emergency, the time evolution of the damaged ship floating position can be exploited instead of floodwater levels (Trincas et al. 2017). In this context, a method has been proposed to correlate the recorded floating position with the main flooding consequences by applying Machine Learning (ML) techniques (Braidotti et al. 2021b). An onboard DSS based on such a technique needs very limited sensors sets: inclinometers (usually fitted on old vessels too) and a measurement of actual draught (which can be obtained from level radar(s) fitted in still-pipes or below bridge wings).

In order to train supervised ML algorithms, a database of progressive flooding simulations in the time-domain is required. The training database is built according to a damage case generator. The present work explores the effect of different damage cases generation algorithms on the prediction of progressive flooding consequences. In particular, two solutions are tested: a Parametric (P) one and one based on Monte Carlo (MC) sampling according to the SOLAS probability distributions for damage dimensions. After a short overview of the progressive flooding consequences prediction, the database generation algorithms are presented. The proposed methodology is applied to a box-shaped barge using a large SOLAS based database for validation purposes.

2 PREDICTION OF MAIN DAMAGE CONSEQUENCES

Considering a flooding casualty, there are a few essential information required to support the master decision. First, the final fate of the ship, means whether the ship will survive the damage scenario reaching a new equilibrium position or will sink, capsize or shift towards an unsafe condition (excessive equilibrium heeling angle). Another important information is the set of flooded watertight compartments, since this knowledge is mandatory to carry out properly the damage control procedures and prevent further spreading of floodwater towards intact watertight compartments. Finally, the time frame of the events is of the utmost importance, especially for non-survival damage scenarios. In such a case, ship abandonment is required, hence, the knowledge of the time-to-flood t_f can help to properly manage the ship evacuation. This basic information can be assessed by employing ML algorithms based on the time evolution of the floating position according to the process depicted in Figure 1, developed in MATLAB.

During the progressive flooding, the vessel floating position changes due to the embarked floodwater. The floodwater spreading is governed by the hydraulic laws, being predictable by applying flooding simulation codes. Hence, a link can be defined between the recorded time evolution of sinkage s, heel ϕ and trim θ angles, used as predictors, and the main flooding consequences, i.e. the responses (Braidotti 2021b). During the emergency, the predictors can be recorded at defined time instants up to t^*, i.e. the current one. Hence their number increase as the progressive flooding evolves. Here, the floating position is recorded at constant time instant dt and, at each time instant, three specific learners are trained to predict the three studied consequences (final fate, flooded compartments and time-to-flood).

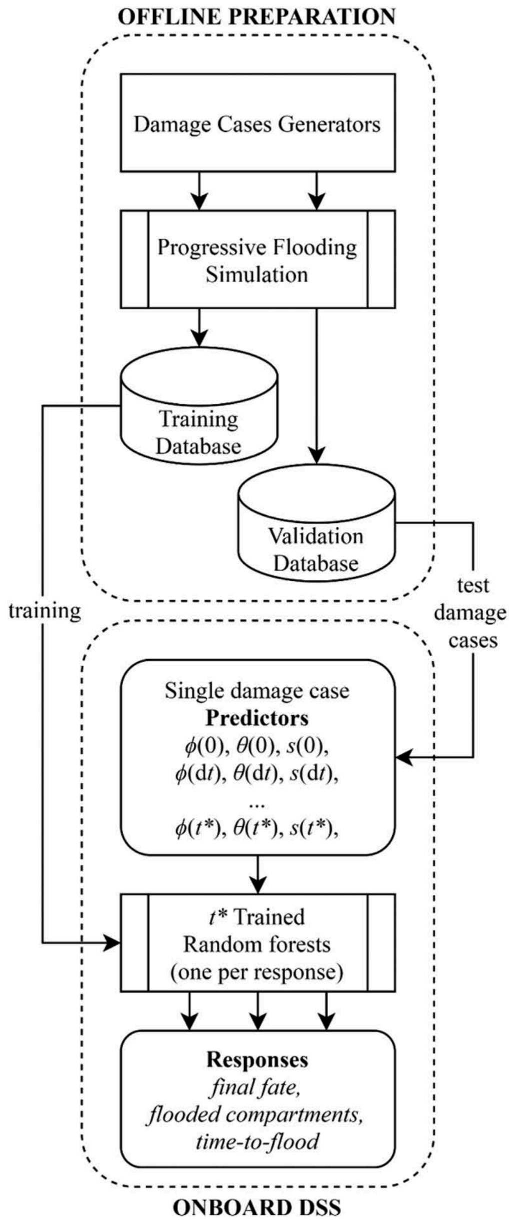

Figure 1. Flowchart showing the preparation of the databases and the process for onboard prediction of damage consequences.

Among the options available in the literature, Random Forests (RF) have shown good performances in addressing the stated problems (Braidotti 2021b). Hence they have been employed in the present work.

These decision trees shall be trained with a database of progressive flooding simulations defined by a proper damage cases generator. Moreover, to validate the process, another independently generated database shall be employed. The progressive flooding simulations included in the validation database provide the predictors' values up to the current instant t^* allowing to statistically test the accuracy of the classification (for final fate and flooded compartments) and regression (for the time-to-flood). Hence, the validation database shall be as far as possible representative of a real probability distribution of damage scenarios.

2.1 *Employed learners*

Decision trees are a popular class of supervised ML algorithms that can be employed in both classification and regression problems (piecewise approximation of the response function). The decision trees are based on binary decisions taken according to the one predictor's value x_i at each node. Hence, the process is shaped like a tree: it starts from a root and decision by decision reaches a leaf, i.e. the response (Figure 2). A single decision tree is trained with a database capable to describe the link between predictors and response (Breiman et al. 1984).

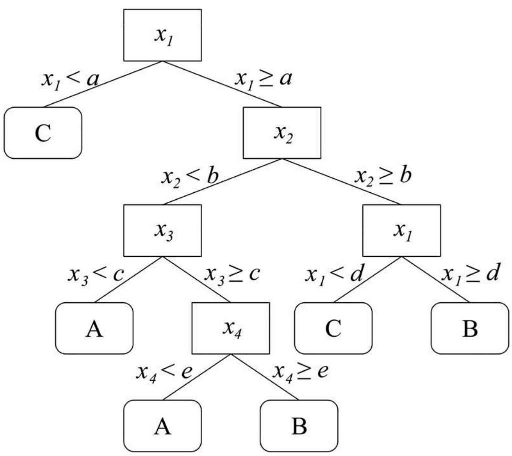

Figure 2. Structure of a decision tree (Braidotti 2021b).

The accuracy of the prediction provided by decision trees can be improved by employing the so-called RF (Breiman 2001). Like bootstrap aggregation, instead of a single tree trained with the complete database, the problem is decomposed in a set of trees trained with a partition of the database. The final response of the overall model is selected according to the vote of the multiples trees for classification problems and as the average of responses for regression ones. Here, 30 weak learners are employed. Besides, to decorrelate the trees in the ensemble, RF employs also a random selection of a predictors' subset each time a split in a tree is considered (James et al. 2017). The RF have proved to be more resilient to noise/missing data and more capable to deal with higher dimensionality data. Hence, the choice of such an ML technique is very suitable for the studied problem involving a large number of predictors for higher t^* and being progressive flooding affected by uncertainties (Rodrigues et al. 2018, Braidotti et al. 2019).

2.2 *Accuracy evaluation*

As mentioned the accuracy evaluation is performed testing the proposed methodology with the damage cases including in a validation database. The performance measure has been carried out with two different approaches for the classification and regression problems, respectively.

Considering classification, the accuracy Acc at a specific time instant t^* is evaluated as the quotient of the number of correctly classified damage cases N_c on the total number of cases within the databases N:

$$Acc\ (\%) = 100\frac{N_c}{N} \tag{1}$$

Aiming to assess the forecast capability of the model, a so-called ongoing accuracy Acc^* has also defined considering only the damage cases having t_f greater than t^*. Means, considering only the N^* scenarios that at that point have not already reached the final stage:

$$Acc^*\ (\%) = 100\frac{N_c^*}{N^*} \tag{2}$$

Regarding the regression problem, the performances are measured employing the coefficient of determination R^2:

$$R^2 = 1 - \frac{SSE}{SS_{tot}} \tag{3}$$

$$SSE = \sum_{i=1}^{N}\left(y_i - y_i^{'}\right)^2 \tag{4}$$

$$SS_{tot} = \sum_{i=1}^{N}(y_i - y_m)^2 \tag{5}$$

Where y_i, y_m and $y_i^{'}$ are the known responses, their mean value and the responses predicted by the model, respectively. Again, besides the overall one, an ongoing coefficient of determination R^2* is defined considering only the $N*$ ongoing damage scenarios.

3 DATABASE GENERATION

In this work, two main methods for database generation have been tested, the first based on MC sampling to define side collision damage cases, the latter based on a parametric generation aimed to cover all the possible damage scenarios involving multiple rooms and compartments.

In both cases, progressive flooding has been simulated using a fast quasi-static code developed in Java (Braidotti & Mauro 2019, Braidotti & Mauro 2020). The employed quasi-static method has deemed appropriate since there is no interest in the dynamic transient simulation. The present section describes the two methods to generate damage cases.

3.1 *Monte Carlo database*

The MC database has been defined according to SOLAS probability distribution. Hence, it is as far as possible representative of realistic side collision damages. SOLAS considers damage as a parallelepiped defined by five parameters: the damage length l_d, the longitudinal position of its centre x_d, the damage penetration b_d, the height of the highest z_{max} and lowest z_{min} damage tip from baseline.

SOLAS provides the probability distribution of the first four parameters (IMO 2018) whereas the distribution related to z_{min} can be retrieved from the literature (Bulian et al. 2019). In the present work, all the internal structures (bulkheads, decks, etc.) are assumed to be intact, hence the damage penetration has been neglected.

Applying MC sampling, randomly generated damage cases can be defined following the so-called non-zonal approach (Kruger & Dankowsky 2019). Considering that the probability distributions have been obtained by statistical analysis of a database of real collision accidents, a large MC database has been always used for validation purposes.

3.2 *Parametric database*

The parametric generation of damages is divided into two phases. In the first, the box-shaped damages are generated considering every single room. In the latter, these damage cases are parametrically combined to define multiple-room damage cases.

Considering a single room, it was observed that the longitudinal position of the damage centre has only a limited impact on progressive flooding (Braidotti et al. 2019). Hence, all the room damages have been applied at half of the room longitudinal extension assuming an l_d equal to the room longitudinal extension. On the contrary, for each room, at least three different vertical positions of damage centre z_d have been considered: at the bottom, half-height, and top. For rooms extending over multiples decks more intermediate positions have been considered corresponding to the main decks' hights from baseline. In order to define damages within room boundaries, the damage centre height shall be corrected for top and bottom damages, considering the applied damage area.

For each possible damage location (x_d, z_d), multiple damage sizes shall be applied. The area of the i-th damage in j-th room, considering the k-th centre is evaluated as:

$$\frac{1}{A_{ijk}} = \frac{1}{A_{max_{jk}}} + \frac{k}{n}\left(\frac{1}{A_{max_{jk}}} - \frac{1}{A_{min_{jk}}}\right) \tag{6}$$

with $k=[1,2,\ldots,n]$, where n is the so-called number of divisions, which is the main parameter governing the database size. Besides, the minimum and maximum damage area (A_{min}, A_{max}) have been also defined. In a real application, floodwater inflow due to very small damages can be taken under control by the bilge system. Hence, for each position, a minimum area can be defined as the one corresponding to an initial inflow equal to bilge pumps capacity. On the other hand, very large damages drive to the almost instantaneous filling of the damaged room. In the present study, the maximum damage area for each room has been estimated for each location as the area that causes the room-filling in 15 s. Furthermore, every single room is assumed as lost at the beginning of progressive flooding (instantaneous flooding) defining an additional damage case.

As the single room damage cases have been defined, they are combined with the ones related to the contiguous rooms. In detail, combined damage cases are elaborated considering all the possible combinations of damage areas of damages having the same centre height (bottom, half-height, top) and sharing a boundary (watertight bulkhead, deck). Hence, at the intersection of a deck and a transverse bulkhead, the combinations are defined considering up to four rooms. In the present work, only one or two compartments damages have been considered. However, the proposed generation technique can be easily extended to a higher number of contiguous compartments.

4 TEST CASE

The proposed methodology has been tested on a box-shaped barge. In the present section, the test geometry is presented, including details regarding the damage cases' generation process applying both the previously described techniques. Then the accuracy of the outcomes of random forests trained with the two databases is compared and discussed.

4.1 *Test arrangement*

The test geometry is a box-shaped barge having a quite complex internal subdivision. It is divided into five watertight compartments and has three main decks. The main rooms in compartments 1 and 3 extend across the first deck from the baseline. Moreover, a longitudinal bulkhead is fitted in some spaces in compartments 1, 2 and 5. The main particulars of the test geometry are provided in Table 1 whereas the internal layout of the barge is provided in Figure 3. More details about the test geometry can be found in Braidotti et al. 2021b.

Table 1. Main particulars of the test barge.

Description	Symbol	Value
Length overall	L_{OA}	75.0 m
Breadth	B	20.0 m
Draught	T	6.0 m
Depth	D	17.5 m
Displacing volume	∇	7500 m^3
Metacentric height	GM	2.685 m

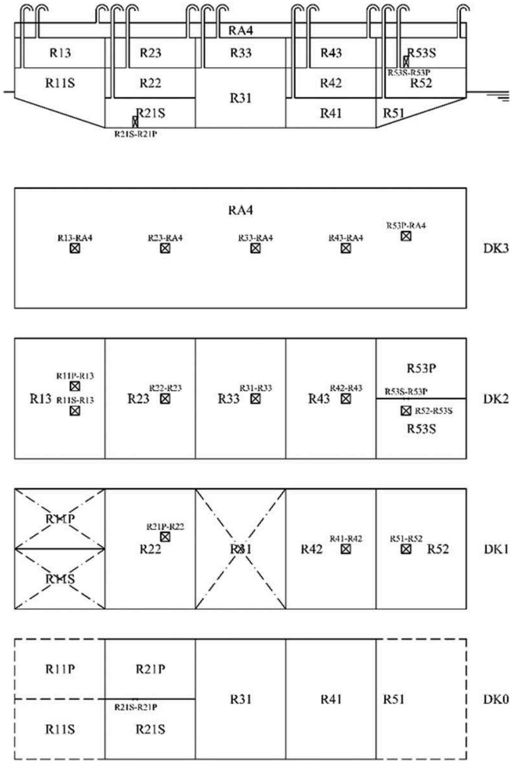

Figure 3. General arrangement of the test barge.

The following databases have been generated:

- MC20: 20,000 damage cases (3,083 non-survival)
- P8: 18,885 damage cases (15,388 non-survival)
- MC50: 50,000 damage cases (8,059 non-survival)

According to previous experiences (Braidotti et al. 2021b), an MC database composed of 20,000 progressive flooding simulations is sufficient to maximize the classification/regression accuracy for the test geometry. Hence, the MC20 database has been assumed for training purposes. In order to assure a fair comparison of the two damage generation techniques, for the parametric database, a subdivision number $n = 8$ has been selected since the resulting P8 database is composed of 18,885 damage cases for the test geometry. Hence the size of the two training databases is comparable. The database MC50 has been assumed as the common validation database in both cases. All the progressive flooding simulations have been carried out up to 2500 s (the limited number of damage cases exceeding such a maximum simulation time are classified as time exceeded).

4.2 *Comparison of the results*

As mentioned, here three problems have been addressed: the classification of final fate, the classification of damaged compartments and the regression of the time-to-flood. In the first and second problem, the random forests have been trained at each time instant t^* with the complete database, whereas, in the latter, only non-survival damage cases have been considered for training and validation. In fact, for survival damage scenarios, the final floating position is reached with a limited exponential trend, leading to a difficult definition of the time-to-flood value. This uncertainty was proved to degrade the performances of the time-to-flood prediction for non-survival cases (for which the definition of the time-to-capsize/time-to-sink is easier since it occurs ad a well-defined instant). It is worth to notice, that the percentage of non-survival damage cases on the total is very different for the parametric and the MC databases. The application of the parametric generation on the test geometry led to 81% of non-survival scenarios while the MC generation according to SOLAS led to 16%.

Figures 4-6 provides the performances of the random forests for all the studied problems. In all the cases the MC20 training database provides better results than the P8 one. This is not very surprising, since the SOLAS probability distribution implies a maximum damage length l_d = 22.72 m, which is larger than the length of one main compartment of the barge. Hence, in MC databases, there is about 7.4% of damage scenarios involving three main compartments. Since the P8 database is limited to damage scenarios involving up to two compartments, cannot properly deal with all the damage cases in the validation database.

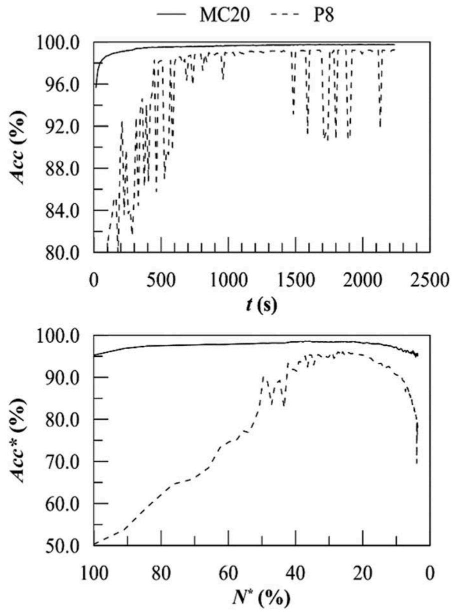

Figure 4. Comparison of classification accuracy in ship final fate prediction.

Nevertheless, it is worth to notice that the gap among the accuracies related to the final fate prediction is about 1% and 3% for overall and ongoing, respectively. So the random forests trained with P8 database are capable to properly classify many damage cases involving three compartments. Confusion matrices reported in Tables 2-5 confirm this behaviour. Anyhow, the results coming from the P8 training database are affected by noise, reach later a stable prediction (larger predictors' set than the MC20 is needed) and show larger decay of the ongoing accuracy as the ongoing damage cases number decreases.

Both the overall and ongoing accuracy in the classification of flooded compartments converges to a gap of about 7%, i.e. the percentage of three compartments damage cases. Moreover, a very strong decay of the ongoing accuracy is observed applying P8 training database, despite it occurs later than the one associated with MC20.

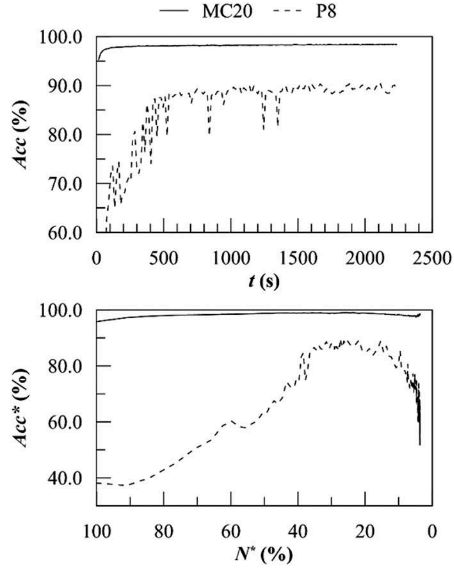

Figure 5. Comparison of classification accuracy in flooded compartments prediction.

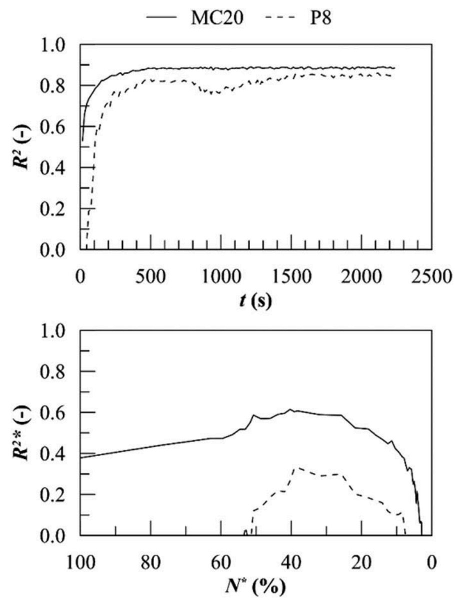

Figure 6. Comparison of regression accuracy in time-to-flood prediction.

Regarding the prediction of the time-to-flood for non-survival damage scenarios, again the results provided in Figure 6 show a better performance of the MC20 database, despite it has a smaller number of non-survival damage cases than the P8 one. The absence of three damaged compartments scenarios in P8 database has

Table 2. Confusion matrix evaluated at 250 s (MC50b damage cases) applying MC20 training database.

	C	S	EH	TE
C	12.8%	0.2%	-	-
S	0.1%	83.0%	-	0.1%
EH	-	-	3.0%	-
TE	-	0.3%	-	4%

C: capsize; S: survival; EH: excessive heeling, TE: time exceeded.

Table 3. Confusion matrix evaluated at 500 s (MC50b damage cases) applying MC20 training database.

	C	S	EH	TE
C	12.9%	0.1%	-	-
S	-	83.0%	-	0.1%
EH	-	-	3.1%	-
TE	-	0.2%	-	0.5%

C: capsize; S: survival; EH: excessive heeling, TE: time exceeded.

Table 4. Confusion matrix evaluated at 250 s (MC50b damage cases) applying P8 training database.

	C	S	EH	TE
C	12.7%	0.3%	-	-
S	12.4%	70.7%	-	-
EH	2.9%	0.1%	0.2%	-
TE	-	0.7%	-	-

C: capsize; S: survival; EH: excessive heeling, TE: time exceeded.

Table 5. Confusion matrix evaluated at 500 s (MC50b damage cases) applying P8 training database.

	C	S	EH	TE
C	12.9%	0.3%	-	-
S	1.0%	82.2%	-	-
EH	-	-	3.1%	-
TE	0.2%	0.5%	-	-

C: capsize; S: survival; EH: excessive heeling, TE: time exceeded.

a strong impact on the ongoing accuracy, leading to a maximum value of R^2* of 0.33 instead fo

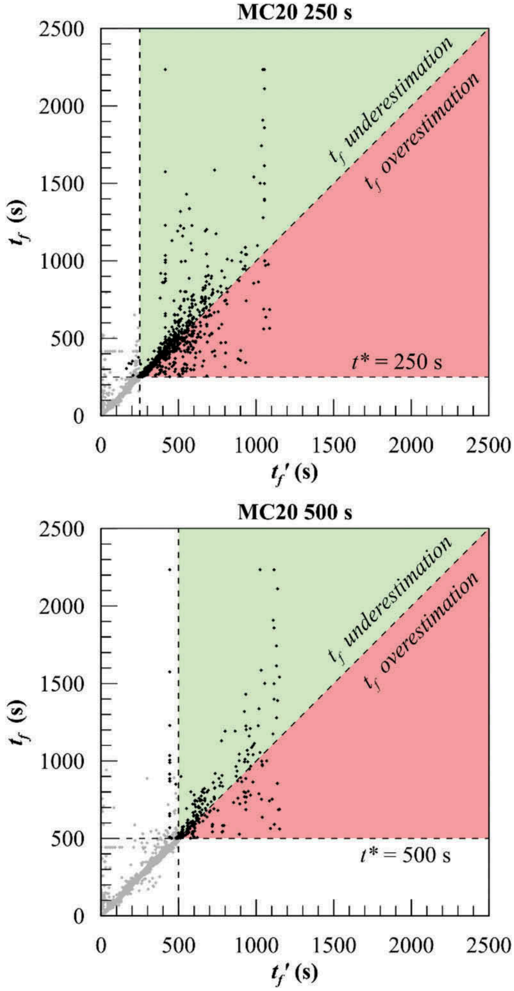

Figure 7. Time-to-flood forecast (MC50b damage cases) applying MC20 training database.

0.61 obtained with MC20 training database, which is anyway quite a small value. In fact, as shown in Figures 7 & 8, the random forests, which provides a piecewise approximation of the response function, are not capable to predict time-to-floods greater than 1200 s applying the MC20 database because of the too reduced number of ongoing damage cases.

The application of the P8 database goes beyond this limitation due to the larger number of non-survival damage scenarios. However, the gain in terms of accuracy is vanished by the strong underestimation/overestimation of the time-to-flood associated with the three damaged compartments scenarios.

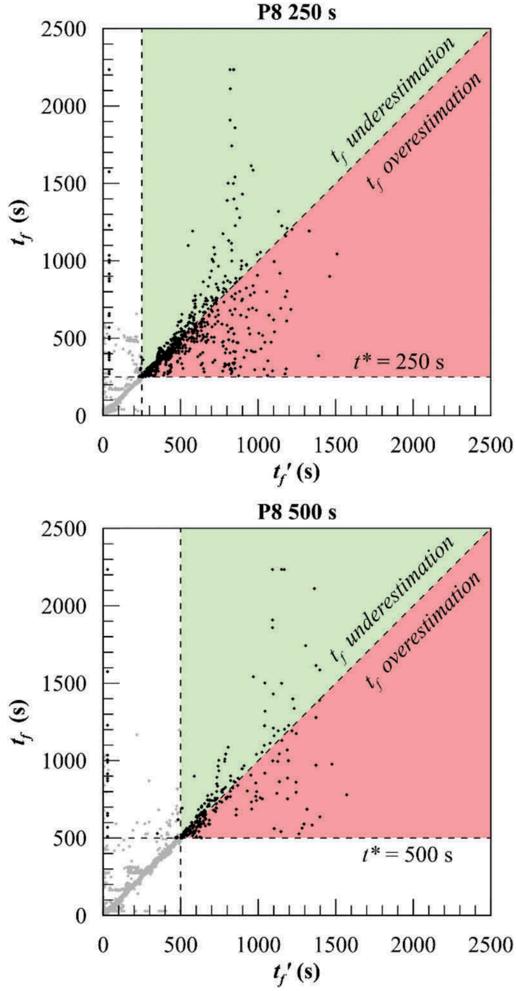

Figure 8. Time-to-flood forecast (MC50b damage cases) applying P8 training database.

5 CONCLUSIONS

This work showed once again the feasibility of the prediction of the damage consequences from the time evolution of the floating position of the damaged ship. Random forests have been applied, obtaining very promising results. Moreover, it has been proved that the process' accuracy is strongly influenced by the adopted training database.

In detail, two options have been explored, one parametric and one based on MC sampling and SOLAS probability distributions. Applying an MC validation database, better performances have been obtained on the test geometry. However, since the parametric training database was limited to two damaged compartments scenarios, a decisive conclusion on which is the preferred option cannot be yet stated. In fact, the parametric database showed quite good resilience for the classification of the ship final fate. Moreover, the parametric generation method led to a higher percentage of non-survival damage cases and thus to a less skewed dataset. Besides, the increased number of long damage scenarios seems to enable extending the prediction capabilities on the time-to-flood beyond the limitations related to an MC training database, which does not predict any time-to-flood above 1200 s. However, further work is advisable to test a parametric database including three damaged compartments scenarios as well as to analyse more complex geometries, such as a full-scale cruise ship. Furthermore, although the RF showed good performances, other ML algorithms, such as neural networks, might be tested or combined in future works to achieve better performances. Finally, the introduction of damage penetration, here neglected, should be as well studied.

ACKNOWLEDGEMENTS

This work has been fully supported by the Croatian Science Foundation under the project IP-2018-01-3739. This work was also supported by the University of Rijeka (project no. uniri-tehnic-18-18 1146 and uniri-tehnic-18-266 6469).

REFERENCES

Breiman, L., Friedman, J., Stone, C.J. & Olshen R.A. 1984. *Classification and Regression Trees*. Boca Raton: CRC Press.

Breiman, L. 2001. Random Forests. *Machine Learning* 45: 5–32.

Braidotti L. & Mauro F. 2019. A New Calculation Technique for Onboard Progressive Flooding Simulation. *Ship Technology Research* 66: 150–162. doi: 10.1080/09377255.2018.1558564.

Braidotti L. & Mauro F. 2020. A Fast Algorithm for Onboard Progressive Flooding Simulation. *Journal of Marine Science and Engineering* 8: 369. doi:10.3390/jmse8050369.

Braidotti, L., Marinò, A. & Bucci, V. On the Effect of Uncertainties on Onboard Progressive Flooding Simulation. In *Proceedings of the 3rd International Conference on Nautical and Maritime Culture-CNM 2019, Naples, 14–15 November 2019*. Amsterdam: IOS pess. doi: 10.3233/PMST190004.

Braidotti, L., Degan, G., Bertagna, S., Bucci, V. & Marinò, A. 2021a. A Comparison of Different Linearized Formulations for Progressive Flooding Simulations in Full-Scale. *Procedia Computer Science* 180: 219–228. doi: 10.1016/j.procs.2021.01.159.

Braidotti L., Valčić, M. & Prpić-Oršić J. 2021b. Exploring a Flooding-Sensors-Agnostic Prediction of the Damage Consequences Based on Machine Learning. *Journal of Marine Science and Engineering* 9: 271. doi: 10.3390/jmse9030271.

Bulian, G., Cardinale, M., Francescutto, A. & Zaraphonitis, G. 2019. Complementing SOLAS damage ship stability framework with a probabilistic description for the extent of collision damage below the waterline. *Ocean Engineering*, 186: 106073. doi: 10.1016/j.oceaneng.2019.05.055.

Dankowski, H. & Krüger, S. 2012. A Fast, Direct Approach for the Simulation of Damage Scenarios in the Time Domain. In *Proceedings of the 11th International Marine Design Conference- IMDC 2012, Glasgow, 11–14 June 2012*.

Hu, L. & Ma, K. 2008. Genetic algorithm-based counter-flooding decision support system for damaged surface warship. *International Shipbuilding Progress* 55, 301–315.

IMO. 2008. *MSC.1/Circ.1291 Guidelines for Flooding Detection Systems on Passenger Ships*. London: International Maritime Organisation.

IMO. 2018. Ch.II-1 Part B Subdivision and stability. *SOLAS 2018 Consolidated Edition*. London: International Maritime Organisation.

James, G., Witten, D., Hastie, T. & Tibshirani, R. 2017. *An Introduction to Statistical Learning: with Applications in R*. New York: Springer.

Kang, H., Choi, J., Yim, G. & Ahn, H. 2017. Time Domain Decision-Making Support Based on Ship Behavior Monitoring and Flooding Simulation Database for On-Board Damage Control. In *Proceedings of the 27th International Ocean and Polar Engineering Conference, San Francisco, 25–30 June 2017*.

Rodrigues, J. & Guedes Soares, C. 2015. A generalized adaptive mesh pressure integration technique applied to progressive flooding of floating bodies in still water. *Ocean Engineering*, 110, 140–151, doi: 10.1016/j.oceaneng.2015.10.002.

Rodrigues, J., Lavrov, A., Hinostroza, M. & Guedes Soares, C. 2018. Experimental and numerical investigation of the partial flooding of a barge model. *Ocean Engineering* 169: 586–603. doi: 10.1016/j.oceaneng.2018.09.042.

Ruponen, P., Larmela, M. & Pennanen, P. 2012. Flooding Prediction Onboard a Damage Ship. In *Proceedings of the 11th International Conference on the Stability of Ships and Ocean Vehicles, Athens, 23–28 September 2012*.

Ruponen, P., Pulkkinen, A. & Laaksonen, J. 2017. A method for breach assessment onboard a damaged passenger ship. *Applied Ocean Research* 64: 236–248. doi: 10.1016/j.apor.2017.01.017.

Ruponen, P., Pennanen, P. & Manderbacka, T. 2019. On the alternative approaches to stability analysis in decision support for damaged passenger ships. *WMU Journal of Maritime Affairs* 18: 477–494. doi: 10.1007/s13437-019-00186-8.

Trincas, G., Braidotti, L. & De Francesco, L. 2017. Risk-Based System to Control Safety Level of Flooded Passenger Ship. *Brodogradnja* 68: 31–60. doi: 10.21278/brod68103.