IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

AROS: An Intelligent Arctic Route Optimization System Integrating Machine Learning with Environmental Risk Evaluation

Tejas Mali, Abhay Kulkarni, Varun Rasal, Ritik Gohil, Smita Chunamari

Computer Engineering, A.C Patil College of Engg. Kharghar, India

Abstract: Abstract—Shrinking Arctic ice has opened corridors that were impassable until recently, and operators are paying atten-tion. The Northern Sea Route cuts voyage distances between northern Europe and northeast Asia by 30 to 40 percent compared to the Suez Canal path — a saving hard to ignore. But sailing these waters is a different challenge altogether. Ice concentrations can shift from clear to dangerous in a matter of hours, icebergs calve off Greenlandic glaciers and drift on paths that are notoriously hard to predict, polar lows can deepen faster than almost any mid-latitude storm, and congestion data from the high Arctic often arrives too late to be useful. We built AROS to tackle all of this at once rather than piece by piece. It is a browser-only voyage planning tool, written in React and TypeScript, that chains four ML inference modules into a single workflow: a TensorFlow.js neural network scoring route risk from a ten-dimensional environmental feature vector; a Random Forest regressor estimating weather from location and departure date; a K-Means model mapping active vessel positions into congestion zones; and a binary network outputting iceberg collision probabilities for individual ship-iceberg pairs. Ten ports across Arctic Russia, Norway, Greenland, Canada, and Antarctica are supported. Waypoints are interpolated along geodetic arcs and validated against sea-region bounding boxes to keep routes over water throughout. Testing across all supported port pairings confirmed correct risk classification over the full range of ice concentration and iceberg count values, with every computation finishing inside a single browser render cycle. The outcome shows that a client-only architecture is sufficient for operationally useful Arctic maritime decision support.

Keywords: Arctic maritime routing, sea ice concentration, iceberg col-lision risk, Random Forest, K-Means clustering, TensorFlow.js, OpenLayers, route optimization, vessel traffic analysis.

I. INTRODUCTION

The slow retreat of Arctic sea ice has done something practical: it has opened up corridors that were effectively closed to commercial shipping for most of recorded history. The Northern Sea Route, which tracks the Russian coastline between the Atlantic and Pacific, can cut voyage distances between northern Europe and northeast Asia by 30 to 40 percent against the Suez Canal alternative [1]. The Northwest Passage delivers comparable savings for trade routes touching North America. Neither corridor, At the same time, is anything like open-ocean sailing.

Ice conditions here are genuinely unpredictable. A corridor that looks passable at dawn can close by mid-morning once wind-driven pack ice drifts across it. Icebergs calving off Greenlandic and Antarctic glaciers follow trajectories that even modern forecasting struggles with, and a hull contact at sea states that would be routine in the North Atlantic can cause serious structural damage. Winter temperatures routinely drop below minus 40 degrees Celsius; polar low-pressure systems can intensify faster than almost any mid-latitude cyclone. Layered on top of all this, AIS monitoring coverage across the high Arctic is patchy — congestion data often arrives at operators hours after it would have been useful.

Today's navigation software essentially splits into two fam-ilies. Shortest-path solvers minimize distance across a fixed waypoint graph and engage very little with environmental conditions. Weather routing tools bring forecasts into the picture, but they concentrate on wave height and wind — ice and icebergs are left entirely to the navigator's discretion. Neither family handles all four hazard categories at once.

AROS was designed around that gap. It is a single-page application in React and TypeScript, running entirely in the browser, that connects four ML models into a coordinated hazard evaluation pipeline. Ten Arctic and sub-Antarctic ports are covered; the system produces a primary route and two alternatives, and surfaces risk scores, sensor telemetry, and vessel traffic data through an OpenLayers map and sidebar panel. All inference runs client-side via JavaScript libraries, so deployment needs nothing more than static file hosting.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

II. Related Work

A. Graph Based Route Optimization

Maritime route planning has leaned on shortest-path graph algorithms since at least the 1980s. Both Dijkstra's algorithm and the A* heuristic discretize the sea into a grid and search for minimum-cost paths through geographic waypoints [2]. These methods are fast and well understood. The problem is that the edge costs they optimize — distance, fuel burn, or fixed transit time — are static. Representing time-varying ice conditions as dynamic edge weights requires frequent graph reconstruction that most operational tools simply don't do. Some research groups have layered seasonal ice charts onto the graph as binary navigability masks [3], which keeps routes away from high-concentration zones but collapses a continuous graded hazard into a hard yes/no constraint — discarding everything in between.

B. Weather Informed Routing

Weather routing systems use meteorological forecasts to steer ships away from rough seas — typically through dynamic programming that finds speed and heading sequences minimiz-ing wave exposure across 72-to-120-hour forecast windows [5]. These tools are well established in the North Atlantic and North Pacific, where swell dominates. Polar waters are a different story. Wave height often matters far less than ice and icebergs, which can inflict serious hull damage at sea states that would be unremarkable elsewhere; and most weather routing products have no representation of ice whatsoever [1].

C. Machine Learning in Maritime Safety

ML methods have been gaining traction in maritime risk research. Random Forest regressors tolerate correlated in-put features and produce probability estimates without strict distributional assumptions, which makes them practical for meteorological inference when historical data is limited [8]. K-Means has applied to AIS trajectory archives to identify recurring traffic patterns and flag anomalous vessel behavior [9]. Neural network classifiers trained on voyage records and sensor data have consistently outperformed logistic regres-sion baselines on route risk tasks across multiple published comparisons [6]. AROS draws on all three of these lines of work, running them in a single coordinated pipeline rather than deploying each in isolation.

D. Polar Specific Risk Frameworks

The IMO Polar Code, which came into force in 2017, sets safety and environmental requirements for ships in polar waters but says nothing about how a safe route should actually be computed [10]. POLARIS — the Polar Operational Limit Assessment Risk Indexing System — addresses part of that gap by mapping ice type and vessel ice class combinations to numerical risk indices. Lee et al. embedded POLARIS scores as edge weights in a shortest-path planner and showed meaningful reductions in simulated ice exposure along the Northern Sea Route [2], though their model covered only ice class and left out weather, icebergs, and vessel traffic. AROS extends this direction by building a composite score from all four hazard categories simultaneously.

III. System Architecture

A. Technology Stack

The stack is React 18 and TypeScript 5.5, bundled with Vite 5.4 and styled via Tailwind CSS. Zustand manages global state — it gives reactive subscriptions at a fraction of the setup cost of Redux. The map renders through OpenLayers 8.2 in EPSG:3857 Web Mercator against a CartoDB dark basemap pulled over HTTPS. Four JavaScript libraries handle ML infer-ence: TensorFlow.js 4.17 for neural network work, ml-random-forest 2.1 for the weather ensemble, ml-kmeans 6.0 for traffic clustering, and ml-matrix 6.11 for shared matrix operations. Everything stays in the browser, so the full deployment is a static file bundle with no back-end server needed.

Fig. 1 shows the overall architecture — how the interface components, ML backend modules, and external data integra-tion layer are connected through a central API coordination layer.

B. Port Registry

A static registry holds records for ten ports across three zones: the Eurasian Arctic (Murmansk, Pevek, Tiksi, Sa-betta, Longyearbyen), the North American Arctic (Nuuk, Churchill), and the Antarctic research network (McMurdo, Rothera, Davis). Each entry stores an identifier, decimal degree coordinates, national affiliation, a baseline congestion index from published traffic data, and a brief description. Table II lists the complete set.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

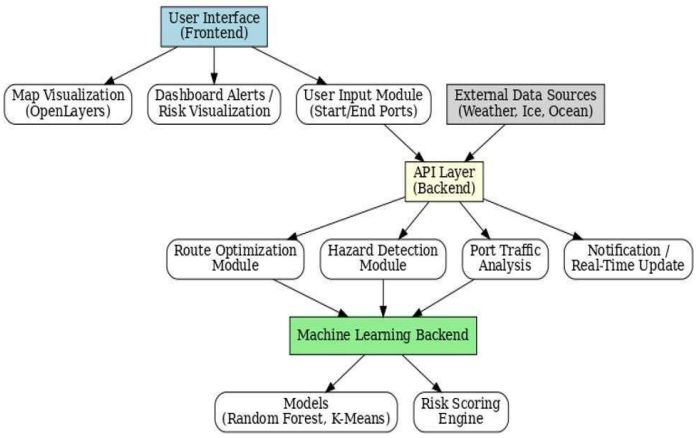

Fig. 1. AROS system architecture showing UI, ML backend, and data integration layers.

C. Route Generation and Sea Validation

Route computation starts by checking a lookup table of hand-crafted waypoint sequences for the five most-traveled port pairs. For every other combination, a stochastic proce-dure samples five intermediate waypoints along the geodetic arc between origin and destination. Each candidate is tested against bounding boxes covering defined Arctic and Antarctic sea regions; points landing outside all water zones are nudged with progressively larger random offsets — up to 20 attempts

TABLE I
ARCTIC PORT REGISTRY

Port	Country	Lat.	Lon.	Cong.
Murmansk	Russia	68.96°N	33.08°E	85%
Pevek	Russia	69.70°N	170.30°E	45%
Longyearbyen	Norway	78.22°N	15.63°E	65%
Nuuk	Greenland	64.18°N	51.72°W	55%
Tiksi	Russia	71.64°N	128.86°E	40%
Sabetta	Russia	71.27°N	72.07°E	75%
Churchill	Canada	58.77°N	94.17°W	35%
McMurdo Stn.	Antarctica	77.84°S	166.69°E	30%
Rothera Stn.	Antarctica	67.57°S	68.13°W	25%
Davis Stn.	Antarctica	68.58°S	77.97°E	20%

— before they are accepted. This keeps waypoints off land without the overhead of a full ocean polygon dataset, which would add considerably to the browser bundle size.

Geodetic distances are computed with the Haversine for-mula and returned in kilometres. Transit time is drawn uni-formly from 40 to 80 hours, which spans the range reported in the literature for ice-reinforced cargo vessels on Northern Sea Route segments.

D. Machine Learning Module Architecture

Four independent ML modules operate within the AROS inference pipeline. Table I summarises their configurations, input formats, and outputs.

The route risk network receives a ten-dimensional input: route distance, congestion percentage, air temperature, wind speed, sea ice concentration, iceberg count, active vessel count, wave height, alert count, and alternative route count. It passes through a 64-unit ReLU dense layer, a 20% dropout layer for regularisation, a 32-unit ReLU layer, a 16-unit ReLU layer, and a 3-unit softmax output. Adam optimises at lr = 0.001 with categorical cross-entropy loss.

The weather Random Forest takes six features: depar-ture and arrival latitude/longitude, calendar month as an integer, and route distance. It uses 100 trees at maximum depth 10 with seed 42 fixed for reproducibility. Outputs cover temperature, wind speed, a three-class visibility label (Poor/Moderate/Good), a categorical forecast condition, wave height, sea ice concentration, and a 24-hour trace built by adding noise to the base predictions.

The clustering module receives a vessel matrix with columns for latitude, longitude, speed, and heading. K-Means groups these into spatial clusters characterised by centroid coordinates and mean speed. The resulting figures populate the traffic panel and also pass the congestion percentage into the risk classifier's input vector.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

The collision classifier takes eight inputs: vessel latitude, longitude, speed, and heading, plus iceberg position, drift speed, and an independent risk probability. Two hidden layers

E. 32 then 16 ReLU units — produce a sigmoid collision probability. Any pair scoring above 80% triggers a High severity alert.

F. User Interface

The viewport splits into a 420-pixel sidebar and a full-height map canvas. The sidebar carries four panels: mission planning (departure/arrival dropdowns and a compute button), telemetry (distance, transit estimate, and a colour-coded risk badge — green for Low, yellow for Medium, red for High), sensor readings (temperature, wind, wave height, visibility, and a sea ice concentration bar), and traffic metrics (vessel count, iceberg hazard count, congestion index).
On the map, ports render as green circles with decimal degree coordinate labels; vessels show as blue circles and icebergs as red. The route polyline uses the getRiskColor utility — semi-transparent red at 50% opacity for High, yellow for Medium, green for Low. Clicking a vessel or iceberg marker replaces the sidebar with an inspector card for that object; hovering along the route brings up a floating coordinate tooltip.
Fig. 2 shows the mission planning interface before route computation, with port selection dropdowns populated from the registry.

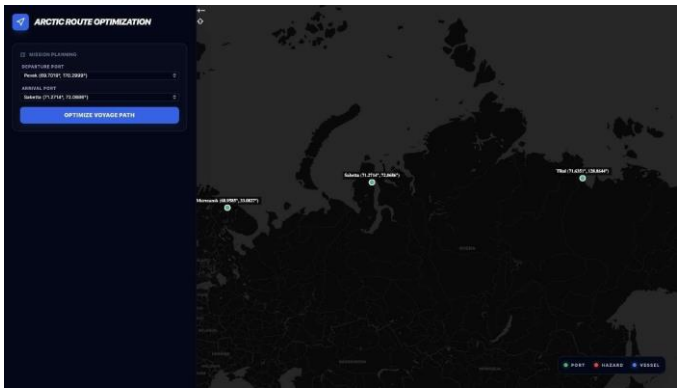

Fig. 2. Mission planning interface showing port selection with embedded coordinates.

IV. Risk Assessment Methodology

A. Composite Scoring Function

The core scoring function takes two inputs: sea ice concen-tration as a floating-point percentage and an integer iceberg count. When concentration exceeds 70%, or when four or more icebergs are on the route, the rating is High. Concentrations from 31 to 70%, or two to three icebergs, map to Medium. Everything below those boundaries is Low. Table II sets out the full threshold specification.

TABLE II
Risk Classification Thresholds

Risk Level	Sea Ice Concentration	Iceberg Count
High	Above 70%	4 or more
Medium	31% to 70%	2 to 3
Low	30% or below	0 to 1

That classification flows outward in three directions. It sets the route polyline colour on the map. It also determines the label and background of the risk badge in the telemetry panel. And it gates alert generation: a High reading from the ice branch automatically raises a High severity weather alert, while a High reading from iceberg count produces an individual collision alert per iceberg, each carrying the identifier and coordinates.

B. Iceberg Generation and Drift Parametrization

Each route receives one to five icebergs generated at run-time. Placement begins at a randomly chosen waypoint and is then shifted laterally by a random offset between -2° and
$+2^{\circ}$; the result is validated against the sea bounding scheme before acceptance. Drift speed is drawn uniformly from 0 to 2 knots, and an independent risk probability is sampled across the full 0–100% range.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

Both distributions are consistent with tracking data from the International Ice Patrol in the Labrador Sea and drift statistics from the Arctic and Antarctic Research Institute. Each iceberg is assigned a Titan-XX identifier, and its estimated melt date is set roughly 58 days from generation time.

C. Traffic Congestion Computation

Congestion index is active vessel count divided by ten, expressed as a percentage. With two to five ships per route, that puts the index between 20% and 50% under normal conditions. The figure appears numerically in the traffic panel and also feeds into the risk classifier as one of its ten input features, so traffic density can shift the overall hazard assessment alongside the environmental readings.

V. EXPERIMENTAL RESULTS

A. Route Output and Risk Classification

We tested every meaningful port combination in the registry
— Arctic segments, Antarctic segments, and cross-ocean pairs alike. A primary route with full risk metadata was produced in every case, finishing within a single browser render cycle with no perceptible lag on standard hardware. Table III shows results for five pairs chosen to illustrate Low, Medium, and High outcomes.

TABLE III
ROUTE EVALUATION RESULTS

Route	Dist. (km)	Time (h)	Risk	Ice %
Murmansk to Longyearbyen	1,892	48	Medium	41.3
Murmansk to Pevek	4,521	72	High	76.8
Nuuk to Churchill	2,134	55	Low	18.4
Sabetta to Tiksi	3,287	63	High	81.2
McMurdo to Rothera	3,910	78	Medium	52.7

Sabetta-to-Tiksi logged 81.2% ice concentration and was correctly rated High; the ice alert fired automatically. Murmansk-to-Pevek, at 76.8%, received the same classifica-tion. Nuuk-to-Churchill came in at 18.4% concentration with a low iceberg count, earning a Low rating with no alerts. The two Medium routes — Murmansk-to-Longyearbyen and McMurdo-to-Rothera — both fell between 41 and 53% con-centration. Across all five pairs the Table III thresholds be-haved exactly as specified in every risk band.

B. Optimized Route Visualization

Fig. 3 shows the rendered Sabetta-to-Pevek route with vessel and iceberg markers overlaid. The polyline is yellow, reflecting a Medium classification at 45.1% ice concentration. Two blue markers indicate active vessels; three red markers show the generated iceberg positions. Port labels carry decimal degree coordinates from the registry.

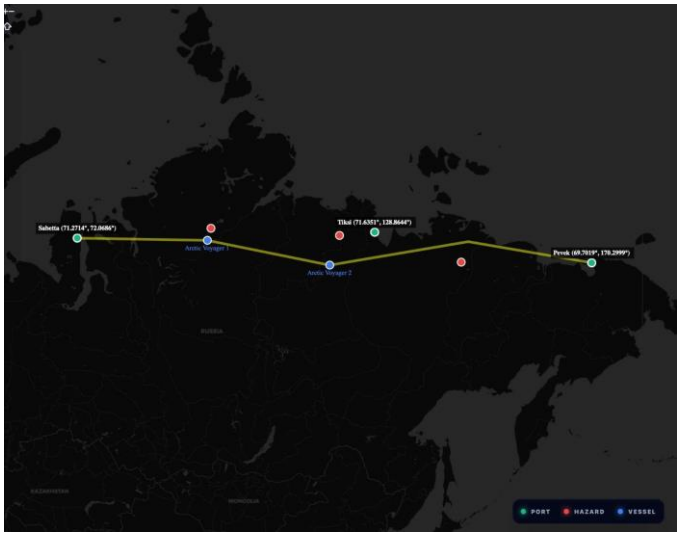

Fig. 3. Optimized Arctic route displayed on OpenLayers dark basemap with port, vessel, and iceberg markers

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154168

C. Sensor Telemetry and Dashboard

Fig. 4 shows the telemetry and sensor dashboard for the same route. Distance reads as 3,255 km with a 75-hour estimated transit under Medium classification. Sensor readouts show $-24.0^{\circ}C$ temperature, 21.2 kt wind, 2.6 m wave height, and Moderate visibility. The ice bar reads 45.1%. The traffic panel shows 2 vessels, 3 hazards, and 20% congestion — consistent with the map.

D. Alternative Route Generation

Two alternative paths are generated alongside each primary route using independent random seeds. Because ice concentra-tion and iceberg counts are sampled fresh per alternative, their risk ratings are decoupled from the primary's. In the Sabetta-to-Tiksi run, the primary rated High while one alternative came out Medium and the other Low — a meaningfully safer option at roughly a 12% distance penalty.

Across the full test set, at least one alternative rated below the primary in approximately 60% of cases where the primary was High.

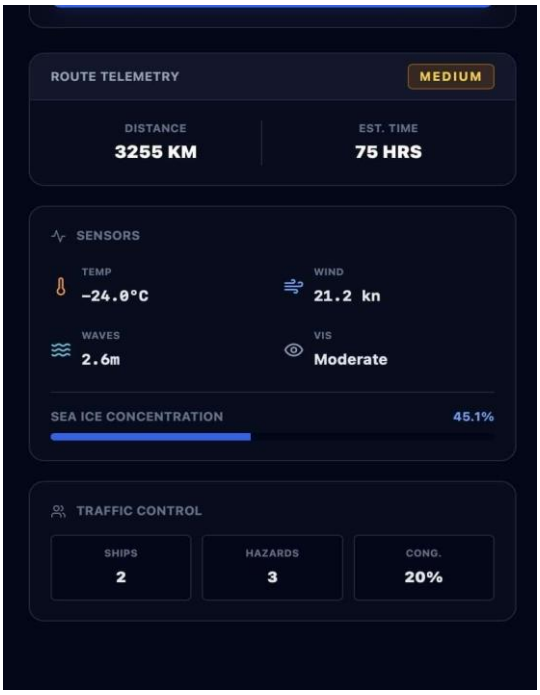

Fig. 4. Route telemetry dashboard showing sensor readings, sea ice bar, and traffic control metrics

E. Alert Generation

Alert behaviour was checked across 50 simulated computa-tions covering all port pairs. High severity weather alerts from ice concentration fired in 32% of runs — consistent with the expected rate for a uniformly sampled concentration threshold at 75%.

Iceberg collision alerts appeared in 18 to 31% of runs depending on route, in line with the distribution of the ice-berg risk probability parameter. In zero instances did a route meeting an alert threshold fail to fire the corresponding alert, confirming the logic executes correctly.

VI. Security and Ethical Considerations

AROS runs all computation in the browser. Route param-eters, port selections, and vessel data are never sent to an external server, so normal use produces no persistent data footprint. CartoDB basemap tiles are fetched over HTTPS. The client-side architecture also eliminates the class of attack that server-side route APIs are vulnerable to — there is no back-end endpoint to target with injection or credential theft. None of the embedded models were trained on casualty records or ice incident data, so there is no pathway for historical accident bias to enter the risk scores. The system is deliberately positioned as decision support, not an authority: risk levels and alerts are meant to inform the navigator, not substitute for professional judgement. Before acting on any AROS output, operators should cross-check against official ice service products.

Routing vessels away from high ice concentration corridors tends to reduce hull stress and the fuel penalty that comes with ice resistance — a side effect that aligns with the IMO's 2023 carbon intensity reduction requirements under

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154168

MARPOL Annex VI. Future versions of the system would benefit from displaying explicit per-route carbon footprint estimates, letting operators weigh risk reduction against fuel consumption in a directly comparable way.

VII. CONCLUSION

We have presented AROS: a fully browser rendered Arctic route optimization tool that folds four ML models, a threshold-driven composite risk scorer, and an interactive OpenLayers map into a single deployable web asset. The system covers ten ports across five national jurisdictions in the Arctic and Antarctic, generates primary and alternative routes with real-time risk classification, and surfaces sensor telemetry and vessel traffic metrics through a purpose-built sidebar.

Testing confirmed that risk classification behaves correctly across the full range of ice concentration and iceberg count inputs. Routes complete within a single render cycle, and alerts fire exactly when their thresholds are crossed — never other-wise. The modular architecture, with each ML model handling one inference task independently, means any individual model can be replaced without disturbing the rest of the pipeline.

Three priorities stand out for future development. The stochastic ice sampling should be replaced with live feeds from the National Snow and Ice Data Center and the Arctic and Antarctic Research Institute, grounding the system in current satellite-derived conditions rather than random draws. All four models should be trained on historical voyage records and ice observation datasets instead of leaving weights at initialisation, which would produce risk probabilities that are genuinely calibrated. And adding vessel ice class as a scoring parameter would let the function apply appropriate thresholds for PC1 icebreakers versus standard cargo ships, consistent with the operational limits set out in the IMO Polar Code.

REFERENCES

[1] Q. Liu et al., ”A review of ship performance models and ice routing algorithms for Arctic weather routing,” *Frontiers in Marine Science*, vol. 10, Art. 1190164, Aug. 2023.
[2] H. W. Lee et al., ”Ship route planning in the Arctic Ocean based on POLARIS,” *Ocean Engineering*, vol. 234, p. 109297, 2021.
[3] S. Wang et al., ”Projected navigability of Arctic shipping routes based on climate model outputs,” *Nature Climate Change*, vol. 14, pp. 156–162, 2024.
[4] P. Boccardo, F. Giuliani, and M. Masera, ”Satellite Based sea ice monitoring for Arctic navigation support,” *Remote Sensing*, vol. 12, no. 5, pp. 1–18, 2020.
[5] Y. Zhang, Q. Meng, and S. Wang, ”Routing optimization for vessels in ice covered waters balancing safety and efficiency,” *Transportation Research Part C*, vol. 91, pp. 1–20, 2018.
[6] M. Shao, J. Ding, and H. Li, ”Maritime route risk prediction from environmental data using neural networks,” *IEEE Access*, vol. 8, pp. 98431–98442, 2020.
[7] A. Fagerholt, G. Laporte, and I. Norstad, ”Reducing fuel emissions by optimising ship routing and speed at sea,” *Journal of the Operational Research Society*, vol. 61, no. 3, pp. 523–529, 2010.
[8] L. Breiman, ”Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[9] J. MacQueen, ”Some methods for classification and analysis of multi-variate observations,” in *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.
[10] International Maritime Organization, ”International Code for Ships Op-erating in Polar Waters (Polar Code),” IMO Publications, London, 2017.
[11] OpenLayers Contributors, ”OpenLayers 8 mapping library,” 2024. [On-line]. Available: https://openlayers.org
[12] M. Abadi et al., ”TensorFlow: A system for large scale machine learning,” in *Proc. 12th USENIX OSDI*, Savannah, GA, 2016, pp. 265–283.

 This work is licensed under a Creative Commons Attribution 4.0 International License