AI assistant for intelligent interaction and route optimisation in offshore turbine maintenance system

Victoria Vysotska[1,†], Kirill Smelyakov[2,*,†], Anastasiya Chupryna[2,†], Maksym Derenskyi[2,*,†], Vadim Repikhov[2,†], and Mykyta Hvozdiev[2,†]

[1] *Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, 79013, Ukraine*
[2] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*

Abstract
The work is devoted to the development and incorporation of an AI assistant that simplifies work with an intelligent software system. The AI assistant can process text information and natural human speech into a set of commands for performing various interactions with the software. The scope of application is an information system for offshore wind turbines' maintenance. The prototype of this system allows the simulation of the planning for offshore wind power turbine maintenance. A module has been developed to generate and visualise optimal routes for a sea vessel to traverse a set of turbines using the shortest path. The ant colony optimisation (ACO) algorithm is used to generate optimal routes. The assistant allows for improvement of the result. The improvement methods used are repeated usage of the ant colony optimisation algorithm on route sections and a trained LLM model. This allows to effectively improve the optimal route, obtaining a higher percentage of optimality and helps the operator in decision-making. The use of artificial intelligence significantly increases the efficiency of interaction with the software.

Keywords
Information system, wind turbine, offshore wind farm, optimal route search, ant colony optimisation, information technology, artificial intelligence, text processing, large language model, intelligent assistant.

1. Introduction

The use of artificial intelligence to assist in decision-making is becoming increasingly popular in various software application areas. It is developing very rapidly due to the widespread use of various large language models (LLMs) for processing text information [1]. At the same time, route planning and dynamic tracking are essential tasks in logistics, which attracts the attention of researchers who continue to conduct various studies in recent decades to obtain more efficient and optimised methods in constructing optimal routes.

Nowadays, the world is actively switching to renewable energy sources, among which a special place has been earned by the production of electricity using offshore wind turbines. Offshore wind turbines are a key technology that allows increasing incorporation of green energy into the electricity market [2].

Typically, wind turbines are situated in groups at a distance from the shore and form offshore power plant farms. When servicing them, sea vessels with crews traverse the group of turbines selected for servicing.

PhD Workshop on Artificial Intelligence in Computer Science at 9th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2025), May 15–16, 2025, Kharkiv, Ukraine
* Corresponding author.
† These authors contributed equally.
victoria.a.vysotska@lpnu.ua (V. Vysotska); kyrylo.smelyakov@nure.ua (K. Smelyakov); anastasiya.chupryna@nure.ua (A. Chupryna); maksym.derenskyi@nure.ua (M. Derenskyi); vadym.repikhov@nure.ua (V. Repikhov); mykyta.hvozdev@nure.ua (M. Hvozdiev)
0000-0001-6417-3689 (V. Vysotska); 0000-0001-9938-5489 (K. Smelyakov); 0000-0003-0394-9900 (A. Chupryna); 0000-0003-1539-8018 (M. Derenskyi); 0000-0002-1274-4205 (V. Repikhov); 0000-0003-4022-9777 (M. Hvozdiev)

Very often, the route is composed manually by the captain of the vessel, who approximately selects the sequence of turbines, which leads to the fact that such routes are often not entirely optimal in terms of such criteria as distance, fuel consumption and travel time. In addition, in such a case, it is difficult to consider the conditions "on the water" and the weather conditions.

Currently, the task of developing automation systems for production and technological processes, including servicing offshore wind turbines, is relevant, mainly to optimise time and other types of resources, as well as to help a person in making decisions during professional activity. The transition to automated calculations of the optimal path for passing between all points requires a powerful mathematical apparatus and a solution to the well-known "Travelling Salesman Problem".

There are many algorithms for solving this problem, among which the following algorithms have been tested:

- Direct traversing of all possible options using brute force
- Random enumeration using the Monte Carlo algorithm
- Ant colony optimisation algorithm

In addition, a relevant task is to develop an intelligent assistant that can interact via human speech and a predetermined set of commands, which this assistant can transform into lower-level interaction with the system. It will allow operators to eliminate the need for long navigation through the software, which will speed up their work. Experiments have shown that with an increase in the number of points among which the optimal route needs to be calculated, the complexity of the problem increases several times, and standard algorithms of direct and random enumeration are not enough since not an optimal amount of resources is spent on covering all the route options. The time required for calculations tends to infinity.

The ant colony optimisation algorithm has shown promising preliminary results with the ability to calculate the optimal route for bypassing fifty points in a short time and one hundred points in a longer, but acceptable time. The result of route generation using the ant colony optimisation algorithm is a path that falls into 90% of optimality, which provides space for the development of various methods and for their improvement. Also, it is possible to face generation artefacts - "loops", which are visualised as intersections of route segments and which should be corrected.

Improvements applied to the software should give a better user experience of interaction with the system, improve the result of generating the optimal route and provide visualisation of the process in the module of the information system for planning the maintenance of offshore wind turbines. The information system should provide a convenient user interface for creating a set of points between which the optimal route is generated, visualising the route generation process to assess results, and gaining a relatively quick understanding of the essence of the process. It is also necessary to develop and integrate an intelligent assistant with which it will be easy to interact.

2. Related works

The problem of planning the maintenance of offshore wind turbines has been considered in various studies. In the paper [2], the authors describe the development of an information system prototype for monitoring and planning the maintenance of offshore wind farms. This system allows for the modelling of the location of turbines, setting maintenance parameters, and optimising routes while considering dynamic weather conditions and resource consumption. However, it does not consider the use of artificial intelligence for task management and optimisation or data detection or computer vision [3].

Maintenance optimisation involves finding the most efficient routes, which is traditionally solved using heuristic algorithms. Paper [4] proposes combining the ant colony optimisation algorithm with the beetle antenna search (BAS) algorithm to improve the search accuracy and convergence rate in route planning. This approach is considered in this paper, and it also uses ACO to calculate the optimal route.

The paper [5] considers the problem of constructing a Pareto-optimal path in stochastic transport networks. This study emphasises the importance of considering dynamic conditions when choosing routes, which is relevant for our approach, which involves adjusting routes by identifying loops and recalculating them.

The paper [1] analysed the effectiveness of modern optical character recognition (OCR) tools, which are vital for automated information processing. These developments can be helpful in implementing an intelligent assistant based on a chatbot, which will manage the maintenance process by interpreting commands in natural language.

Modern research also shows the potential of large language models (LLMs) in improving user interaction with automated systems. Papers [6, 7] consider the use of LLM for user support and software version control, which confirms their effectiveness in converting text queries into structured commands. These approaches can be helpful for implementing an intelligent assistant in a maintenance management system. Also, the papers [8] study the use of the LLM model design approach for natural text processing. In the paper [9], alternative methods for solving the shortest path problem using integer residual arithmetic were considered. They were also regarded as methods of data classification described in the paper [10].

The planned assistant should process text information, translating natural human speech into commands and executing them for preparing the maintenance for offshore wind turbines, and be able to improve possible route generation artefacts, "loops", improve the route itself, bringing it closer to the most optimal result. In paper [11], the authors suggested an approach for processing natural human speech.

Although this approach differs from heuristic algorithms, it demonstrates the possibility of optimising computational processes in routing problems. Thus, the system we propose combines an intelligent assistant with an offshore wind turbine maintenance optimisation system, using ACO for initial route planning and its subsequent adjustment by identifying loops. This approach improves the efficiency and adaptability of the offshore wind turbine maintenance process.

3. Methods and materials

Efficient planning of offshore wind turbine maintenance requires an integrated approach that considers many factors. Among them, the key ones are the location of turbines, weather conditions, logistical constraints, characteristics of the vessels performing the maintenance, and the time frame for inspections. Given the complexity and dynamism of the task, a system based on a combination of an ant colony optimisation algorithm for calculating the optimal route, a mechanism for improving its results with the solving of the so-called "loops", and an intelligent assistant based on LLM, which provides operators with support in decision-making process, was proposed, which is also discussed in papers [12, 13].

3.1. Data for optimisation

Before planning routes, it is necessary to collect and analyse data that influences the entire process. The following are considered as initial parameters:

- Coordinates of wind turbines requiring maintenance
- Current and forecast weather conditions
- Technical characteristics of vessels used for maintenance, including their speed and fuel consumption

It is also necessary to analyse the turbine maintenance process. It was considered in [14]. These data represent a dynamic system changing in real time. Therefore, it is necessary to use algorithms that can adapt to new conditions and adjust the route when the input parameters change. When collecting data, the capabilities of the LLM model were also used, as in [15]. Papers [16-17] explore storing data approaches that are crucial to allow efficient processing of data from sensors and weather data.

3.2. Ant colony optimisation algorithm

The ant colony optimisation algorithm is used to build a practical route, and it is widely used in combinatorial optimisation problems. The basic principle of the algorithm is that virtual agents, called "ants", imitate the behaviour of real ants, moving along possible routes and leaving traces - pheromones. These traces affect subsequent iterations of the algorithm, increasing the probability of choosing the most effective paths. The algorithm's operation process includes several stages:

- Formation of a route graph, where nodes represent wind turbines and ports, and edges are possible paths between them
- Launching virtual agents that move along the graph, choosing routes based on a probabilistic mechanism
- Leaving a pheromone that is amplified on the most popular routes, which helps to find the optimal solution
- Iteratively refining the route based on accumulated data

However, despite the effectiveness of ACO, the algorithm does not always eliminate the formation of non-optimal sections of the path, such as loops or return movements. Therefore, a route optimisation mechanism was additionally implemented.

3.3. Information system model

Software was developed to allow the wind turbines to be located on a sea surface map and provide turbine maintenance planning for the experiments. In the process of arranging the maintenance, the operator selects the turbines on which the ship will be carried out and chooses the vessel that goes out to sea for maintenance. The information system was developed using Web technologies according to the principles of "client-server" architecture. The simulator software architecture is shown in Figure 1.

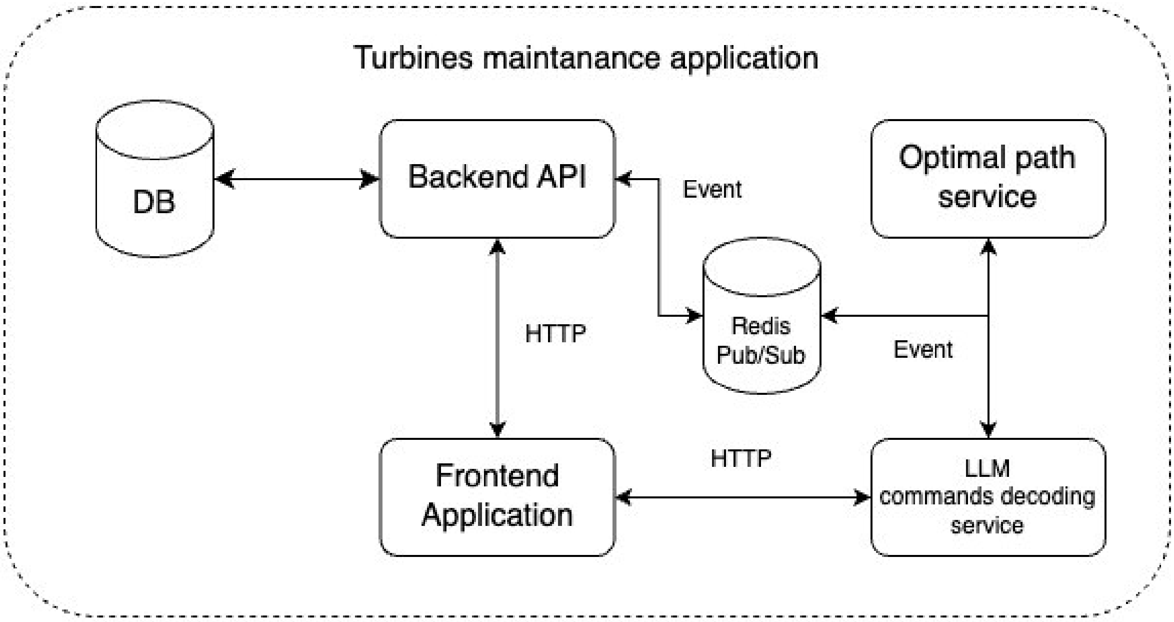

Figure 1: Software architecture

Turbine Maintenance Organisation Application is a general name for a group of services that provide functionality for servicing offshore power plant turbines and assistance in decision-making when organising maintenance. The application is a set of services built on a microservice architecture based on the works [18-20] and consists of the following elements:

- Frontend Application - user interface
- Backend API - server part of the system
- DB – database
- Optimal path service - service implementing the logic of generating the optimal path and applying various improvements to the results
- LLM commands decoding service - a service for processing text information using a large text model

The backend API is built as a single API, divided into logical parts, resources for processing entities, organising and managing turbine maintenance, and interaction with other services. The Node.js runtime environment is used for the application. The code is written using TypeScript, which is an add-on to JavaScript and has strong typing. The Express library is used, which provides components for basic server logic. The PostgreSQL relational database management system is used as a database. A data protection system is also implemented based on examples from paper [21].

The interface of the Frontend Application editor page is shown in Figure 2.

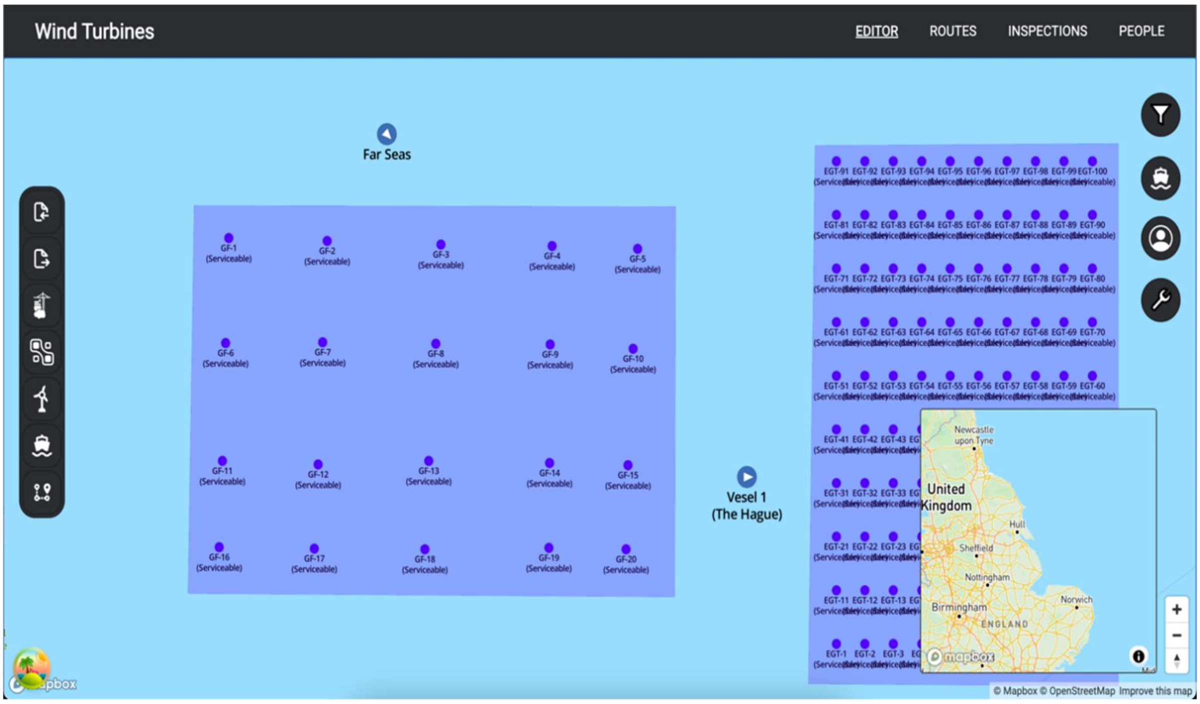

Figure 2: Frontend application interface

For the user, the interface is a map of the Earth's surface, which displays various entities, farms, and offshore wind turbines. Operators can manage entities using controls. The application can store information locally to optimise queries, which was implemented based on the works [22, 23]. During development, the best practices shown in [24] were used.

On the entity editor page, the operator can place entities on the map using the entity side menu. The entity editor is needed to enter initial information about ports, farms, turbines, ships, and teams into the system so that the system has data that can be worked with. Since this is a decision support system, all data must be as close to reality as possible for further experiments to be successful.

Operators enter information about the location of ports, farms, wind turbines, and ships. Next, it is necessary to organise an inspection to service wind turbines. When creating a new inspection, it is essential to select a list of turbines for servicing, as shown in Figure 3, which will later become points for calculating the optimal route between them.

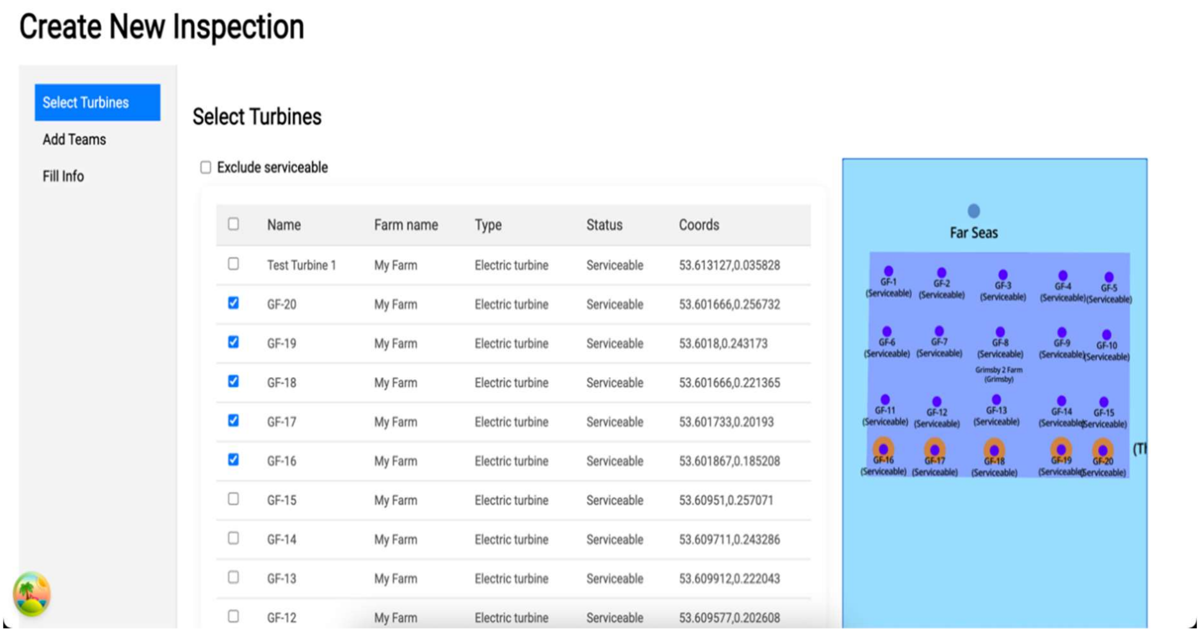

Figure 3: Selecting offshore turbines for inspection

Next, the operator selects an inspection team, as shown in Figure 4 and selects inspection dates as shown in Figure 5. Also, a vessel is selected for inspection, which will traverse all the offshore turbines chosen for inspection. At this stage, it is possible to form new teams, edit and disband existing ones, and select different groups of teams such as the ship's team, the personnel servicing the marine turbines, and others.

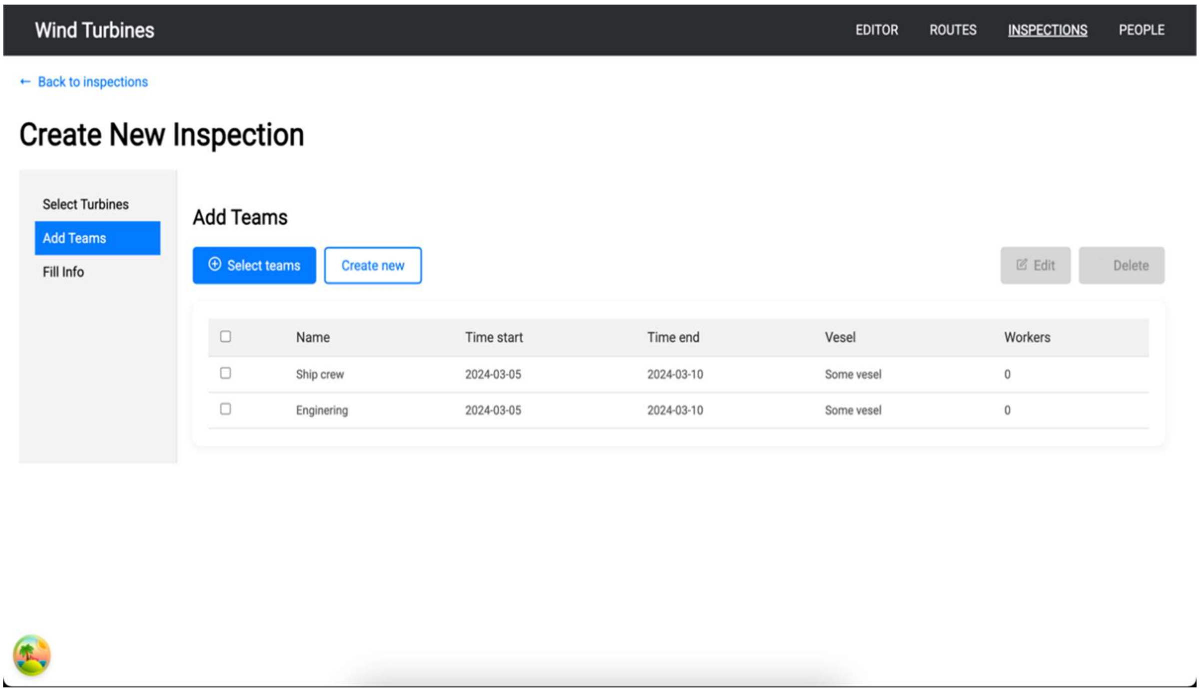

Figure 4: Selecting teams for inspection

The operator can edit an existing inspection by adding or removing turbines from the inspection, edit the inspection commands and dates, or deleting the inspection. When an inspection is opened by the operator or as expected by the captain on the vessel, the inspection screen is presented.

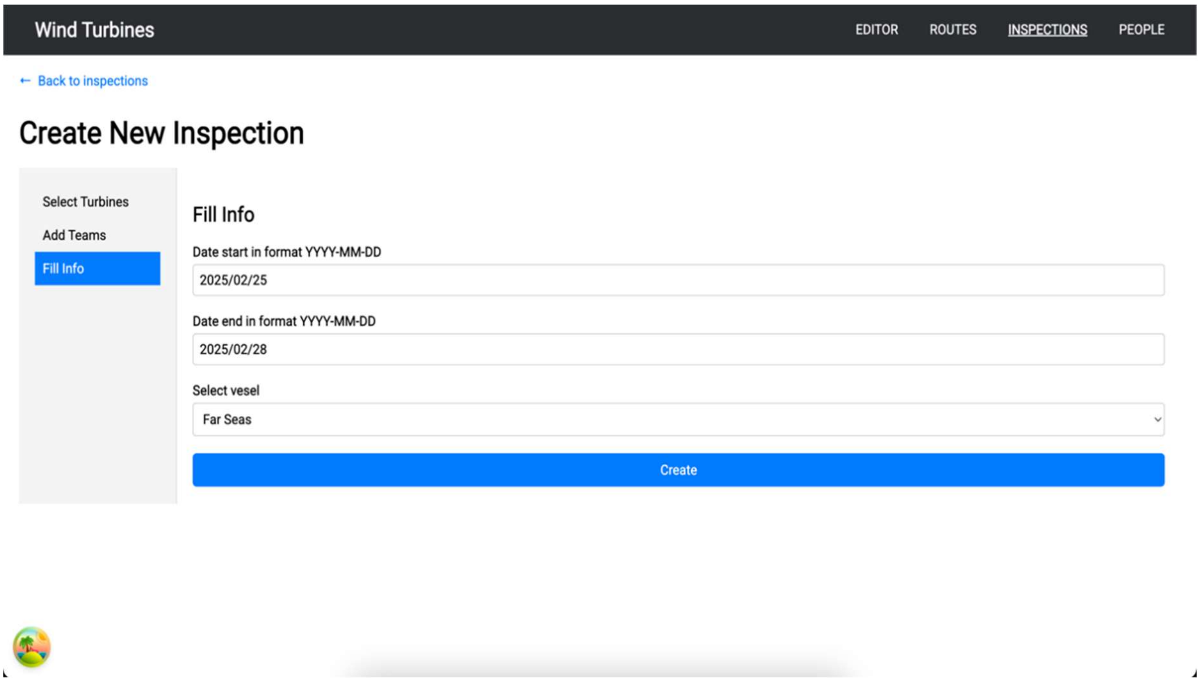

Figure 5: Selecting inspection dates

The inspection screen shown in Figure 6 is a map of the Earth's surface that displays the farm containing the wind turbines selected for inspection, the turbines themselves, and the ship that will deliver the team to the turbines. An optimal inspection route is also automatically generated, showing the expected optimal path to bypass all the turbines based on the criterion of the shortest distance.

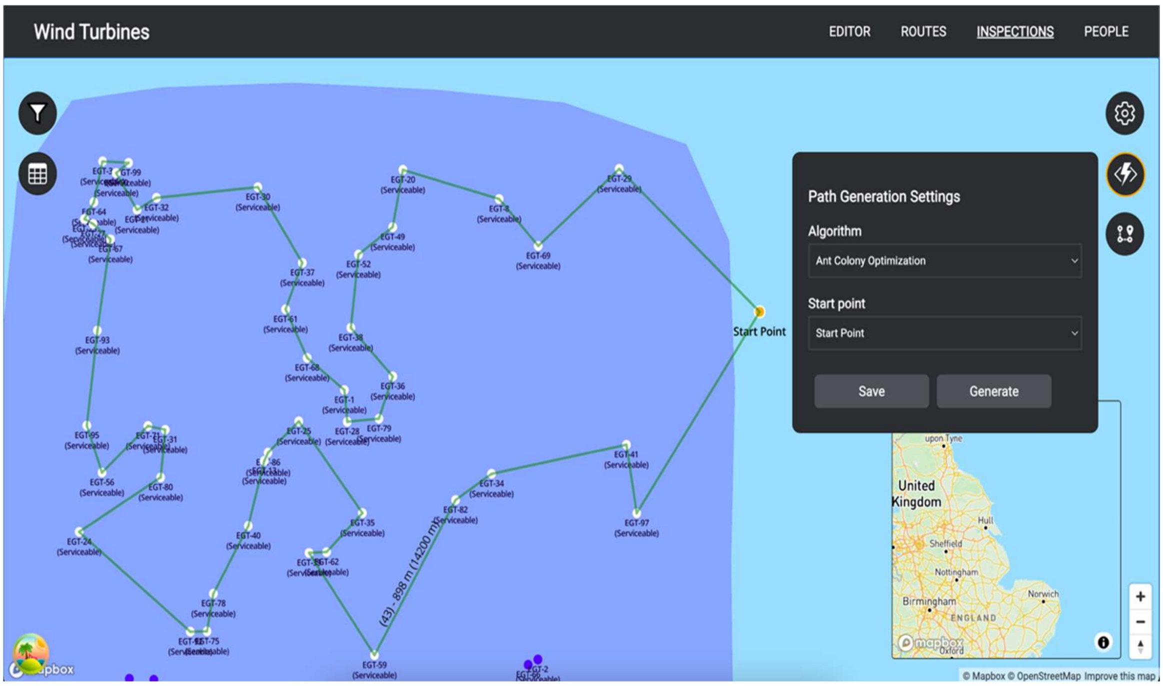

Figure 6: Inspection screen

The operator can view and change the parameters for generating the optimal path by selecting from a list of algorithms for calculating the optimal path in the user interface element shown in Figure 7. The algorithms include:

- Direct traversing of all possible options using brute force
- Random enumeration using the Monte Carlo algorithm
- Ant colony optimisation algorithm

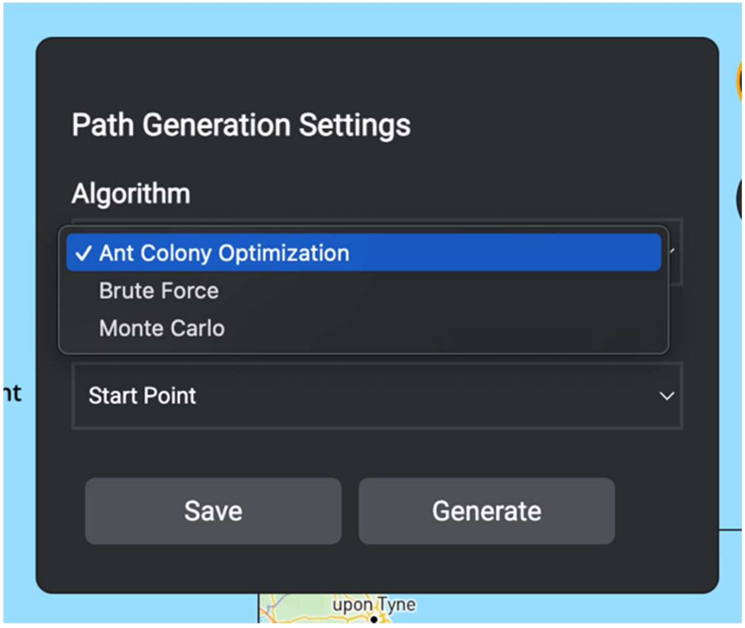

Figure 7: Selecting an algorithm for calculating the optimal route

For the convenience of conducting experiments, it is possible to change the list of turbines selected for inspection, and therefore the points for generating the optimal route. The functionality is shown in Figure 8.

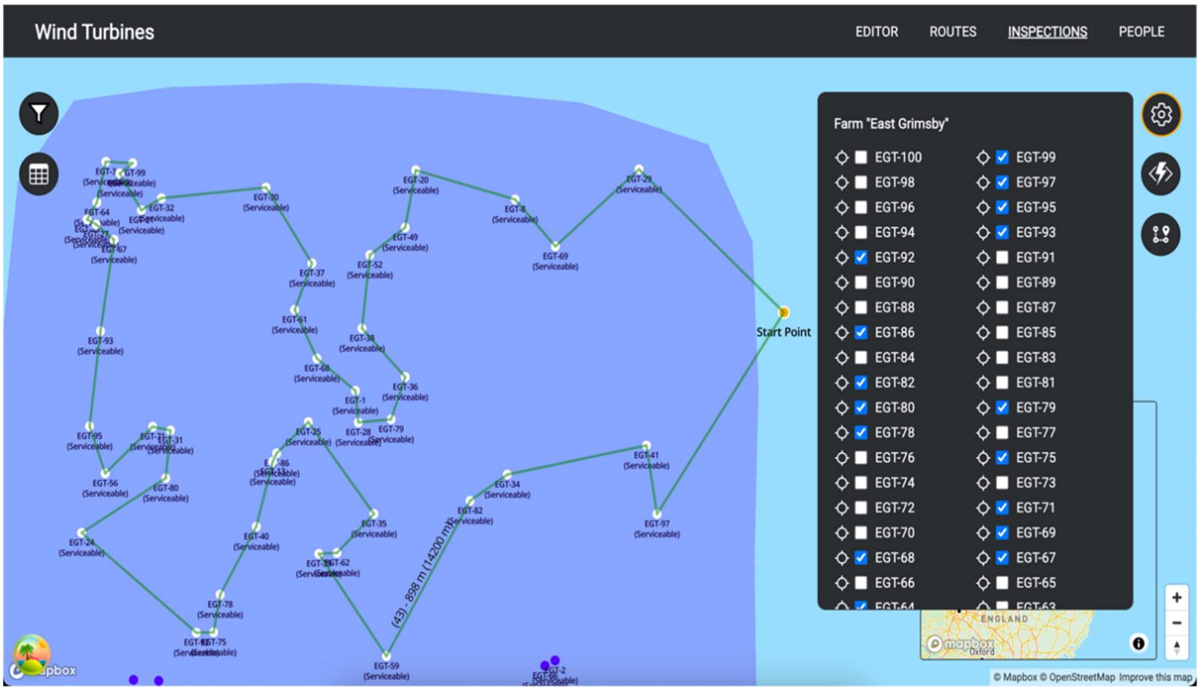

Figure 8: Changing the list of turbines for inspection

Once the optimal path is calculated and displayed, the operator has an opportunity to view it and, if necessary, make changes to the turbine traversing order manually. Using the interface shown in Figure 9, you can move the chips with the turbine name depending on the position in which they need to be traversed. The traverse positions are indicated by a number to the left of the chip with the turbine name.

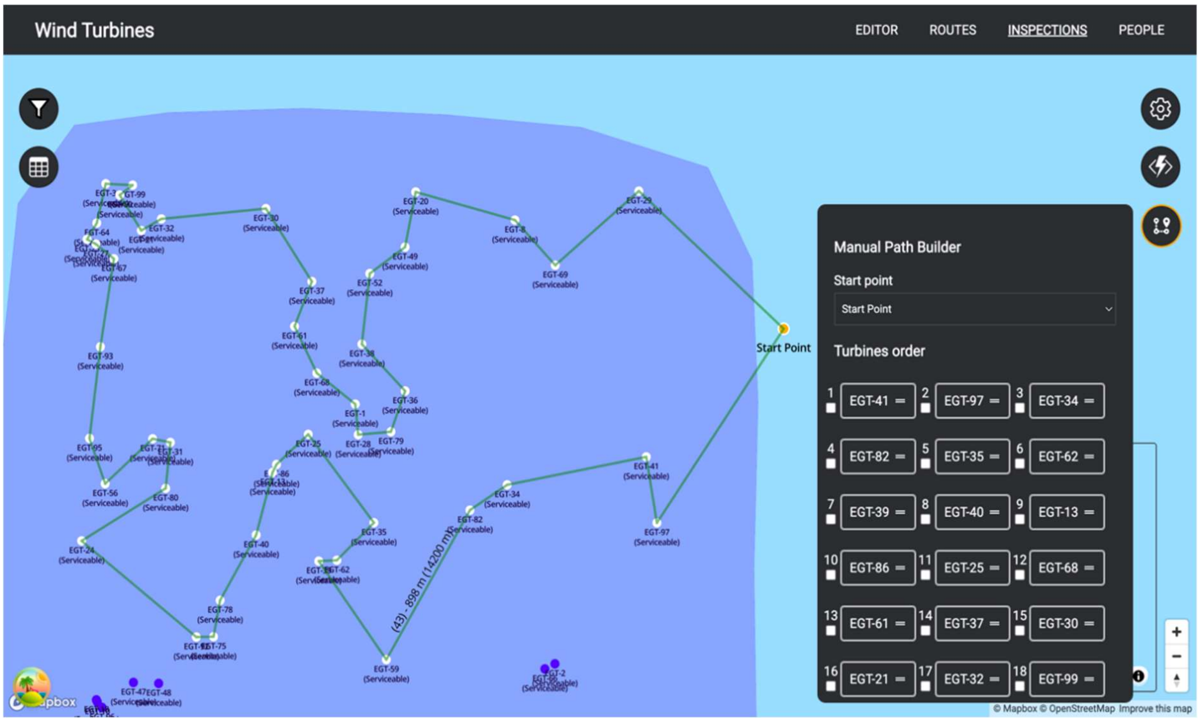

Figure 9: Interface for changing the turbine traverse order

If artefacts or "loops" are detected during the generation of the optimal path, the operator should optimise the path manually. Using the interface described above, the operator changes the order of bypassing the turbines. An example of resolving "loops" is shown in Figure 10.

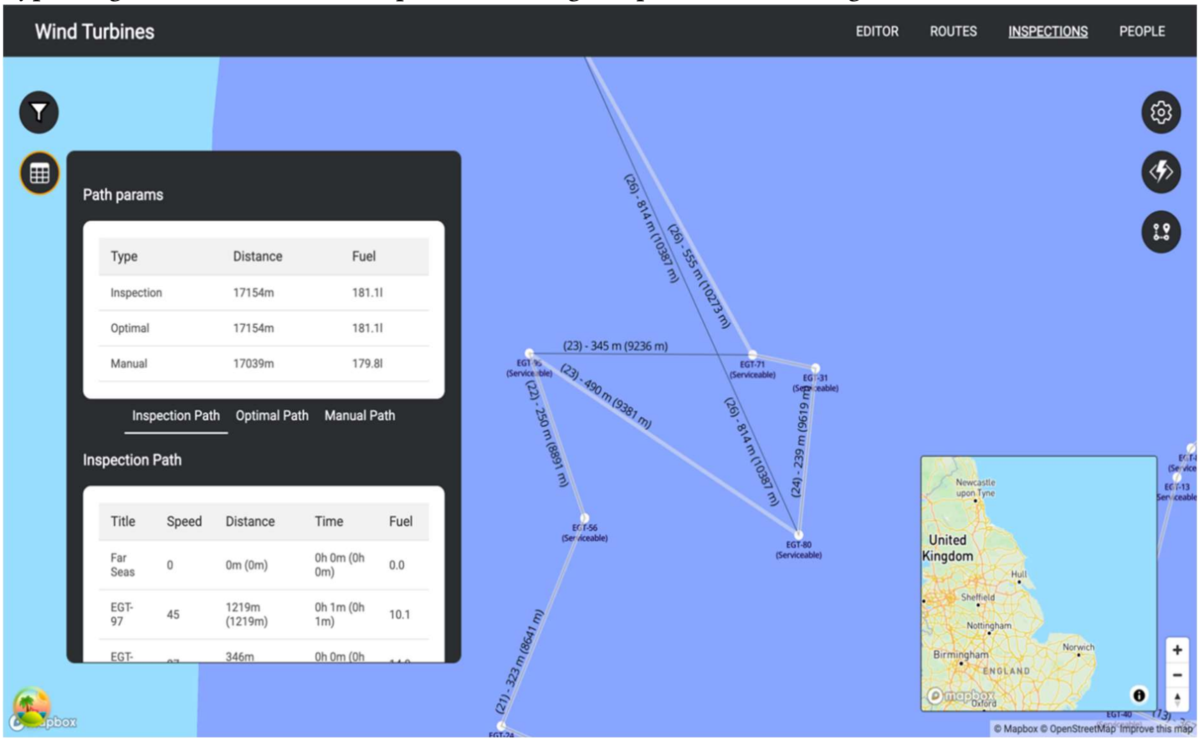

Figure 10: Manual "loop" optimisation

The "loops" form local non-optimal sections of the route, which move the entire path away from the optimal route. In the example, the path shown as a black line forms a loop, and its total distance is 17154 meters. The path shown as a white line is manually optimised. Its total distance has become smaller and is 17039 meters.

3.4. AI assistant

One of the key components of the proposed system is an intelligent assistant designed to assist operators in the planning and decision-making process. The primary purpose of the assistant is to automate the processing of text commands and interaction with the system based on natural language. It performs the following functions:

- Analysis of operator text messages and their transformation into software commands
- Simplification of operator work by reducing manual operations
- Automatic route correction

Integration of the assistant allows for reducing the workload of the operator, speeding up the decision-making process and reducing the probability of errors.

In the future, the system can be expanded by integrating the processing of audio commands and text information using LLM. It will allow:

- Recognise and interpret operator voice commands
- Automatically analyse and structure maintenance reports
- Improve interaction with the system, making it more intuitive

Thus, the proposed system combines advanced routing algorithms, dynamic optimisation mechanisms and modern artificial intelligence technologies. It allows for significant improvement in the efficiency of offshore wind turbine maintenance, reducing operating costs and simplifying the work of operators. Let's examine the principles of the operation of the intelligent assistant. A smart assistant is necessary to reduce the time it takes for the system operator to perform basic actions in the system and to receive assistance in decision-making. An example is the creation of a new inspection. The operator can interact with the assistant in human text format, sending it a message with a request to perform some action. Figure 11 shows a request to create a new inspection.

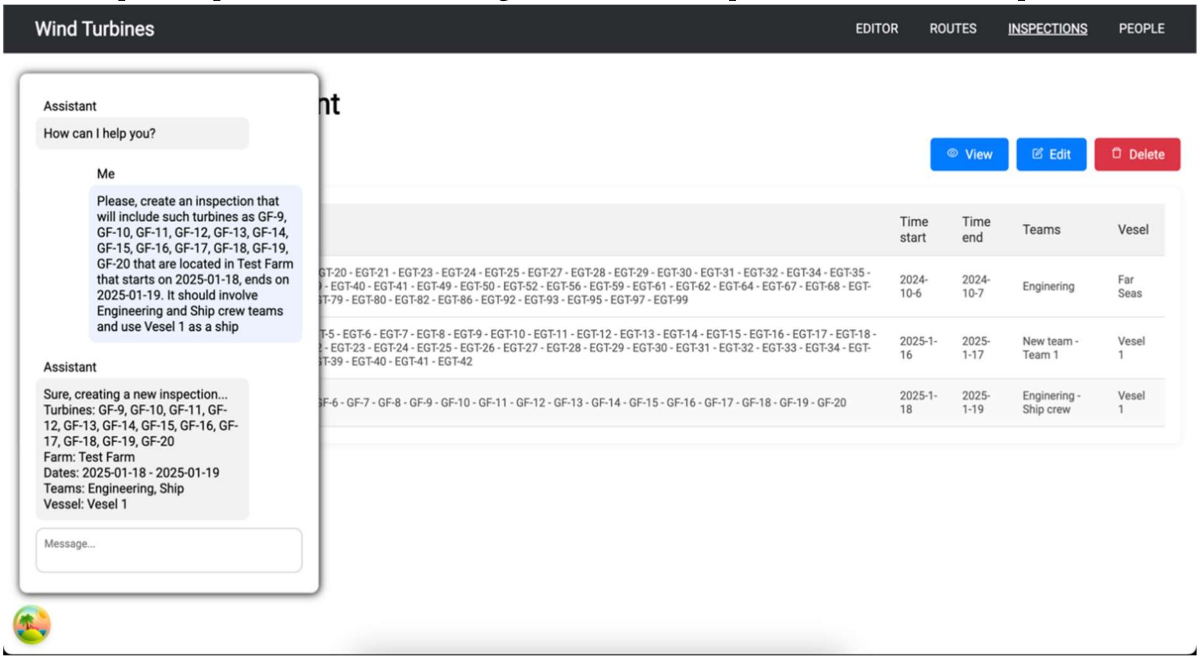

Figure 11: AI assistant interface

The language model is trained to understand the user's query and match it with the command keys and arguments that are then returned, and based on which the program performs specific actions. For example, the text query "Please, create an inspection that will include such turbines as GF-9, GF-10, GF-11, GF-12, GF-13, GF-14, GF-15, GF-16, GF-17, GF-18, GF-19, GF-20 that are located in Test Farm that starts on 2025-01-18, ends on 2025-01-19. It should involve Engineering and Ship crew teams and use Vesel 1 as a ship, which will be converted to a command key with the arguments shown in Table 1.

Table 1
Converting a text query to a command

Part of the query	Command	Arguments
Please create an inspection that starts on 2025-01-18 and ends on 2025-01-19, and use Vesel 1 as a ship.	createNewInspection	vessel: Vesel 1 startDate: 2025-01-18 endDate: 2025-01-19
include such turbines as GF-9, GF-10, GF-11, GF-12, GF-13, GF-14, GF-15, GF-16, GF-17, GF-18, GF-19, GF-20	assignTurbines	turbines: [GF-9, GF-10, GF-11, GF-12, GF-13, GF-14, GF-15, GF-16, GF-17, GF-18, GF-19, GF-20]
It should involve the Engineering and Ship crew teams	assignTeams	teams: [Engineering, Ship]

In the sequence diagram shown in Figure 12, you can see that the following actions occur: the operator enters a message asking to create an inspection and describes its parameters, such as the turbines to be inspected, inspection dates, ship, and commands. This text is sent to the LLM Decoder service, where the text model converts the text into a command key and receives the necessary arguments. Then an event is sent, which is caught in the Backend API, and the essential command is executed with the passed arguments. After successful execution, the new inspection is displayed in the client application.

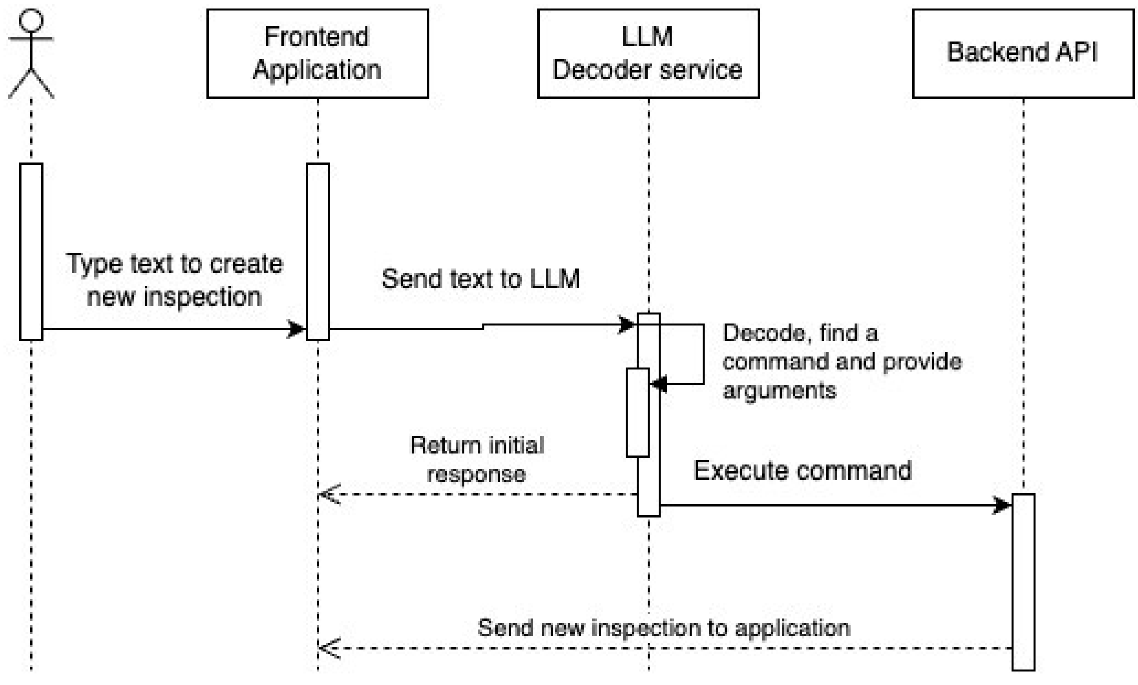

Figure 12: The process of creating an inspection with an AI assistant

Processing natural language and text information is an auspicious direction in software development. The works [25, 26] show some practices of using LLM models for processing and creating recommendations, protecting data and helping the user make decisions. In the future, it is also possible to implement support for converting voice to text, as is being done [27, 28], which will allow the system operator to give commands using voice, which will speed up interaction with the system many times over. The most promising results are suggested in the paper [29], where the authors used the LLaMa model.

4. Experiment

The main way to conduct experiments in the decision-making information system is to simulate the real work of the operator. The real work of the operator includes creating an inspection, setting its parameters, turning on and managing the turbines that need to be inspected, adding commands, choosing dates and a ship. The experiment will also consist of further management of the inspection, which includes setting the starting point of the inspection and building an optimal route to bypass the turbines using the ant algorithm, further improving the result of generating the optimal route. It should bring the simulation as close to reality as possible.Several experiments are planned to demonstrate the work of the information system. The following experiments are planned:

- An inspection is created by the operator with the help of the AI assistant.
- Setting the inspection starting point manually and with the help of the AI assistant.
- Generating an optimal path using the ant algorithm.
- Improving the result by recalculating sections with intersections.
- Improving the result with the help of the AI assistant.
- Comparing the results.

The process of creating an inspection has already been described above; it consists of selecting the marine turbines that need maintenance, building teams for maintenance, setting maintenance dates and choosing a ship. Creating an inspection by the operator using the AI assistant reduces the number of actions required to write only one message. The second experiment consists of setting a starting point in front of the farm, from which the vessel begins to traverse selected turbines and returns to it. Also, calculations of the optimal path perceive this point as the start and finish. In normal mode, the operator must set this point on the map manually, as shown in Figure 13.

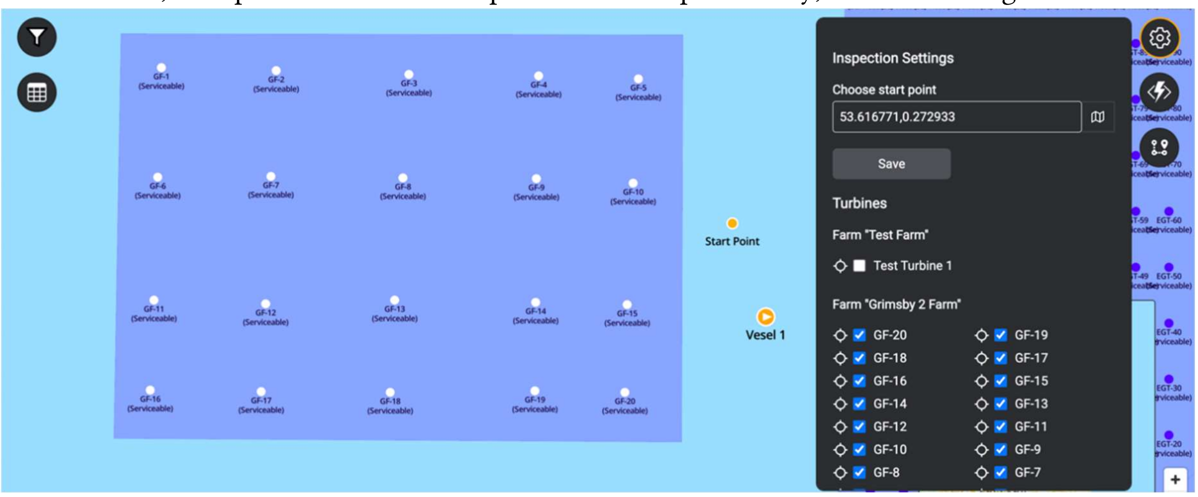

Figure 13: Setting the starting point manually

With the AI assistant, the entire process also comes down to writing a text message in which the operator specifies the coordinates of the starting point, as in Figure 14. The conversion of text into a command key and arguments is given in Table 2.

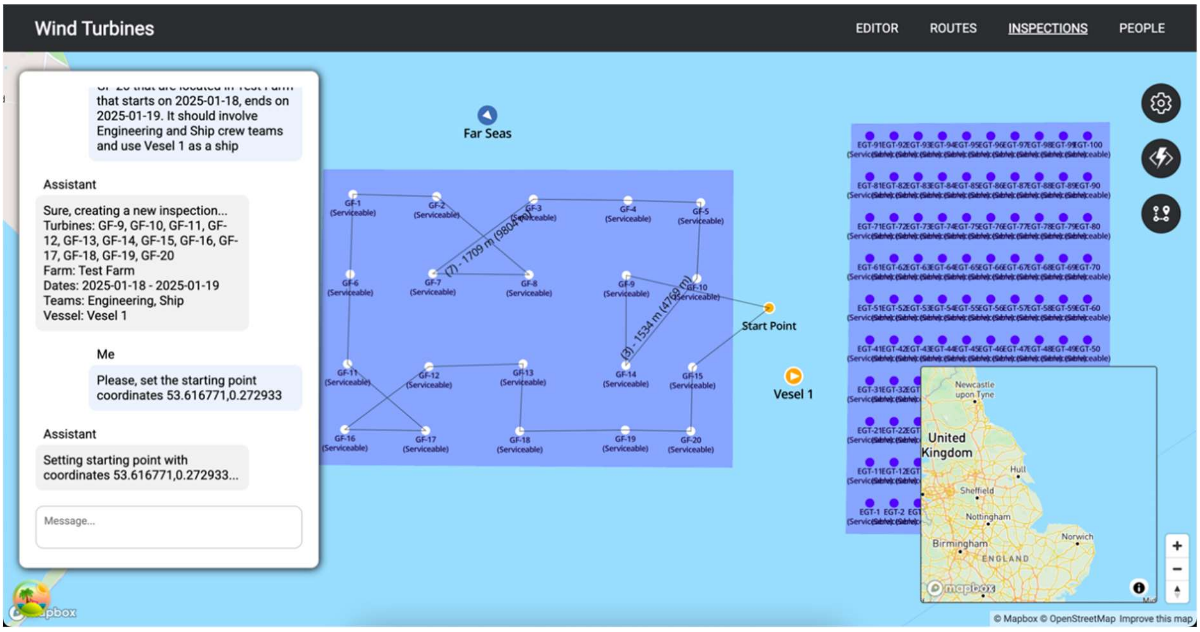

Figure 14: Setting a starting point using an AI assistant

Table 2
Converting a text query to a command

Part of the query	Command	Arguments
Please, set the starting point coordinates 53.616771,0.272933	setStartingPoint	coordinates: [53.616771,0.272933]

Generating an optimal path is the most essential part of the module of the software system for servicing marine turbines. The optimal route is calculated using the ant algorithm. The ant algorithm allows finding the optimal solution to the Komi Voyager problem for a larger number of points. When opening an inspection, the optimal route is calculated automatically. An example of calculating a path for 20 points is shown in Figure 15.

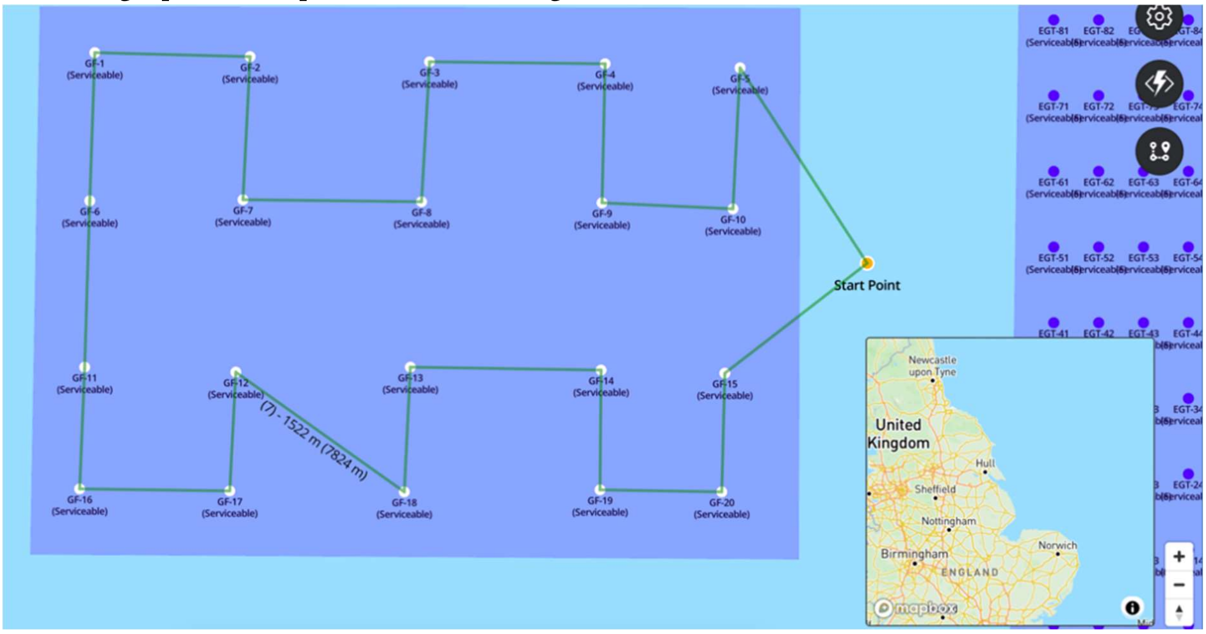

Figure 15: Result of calculating the optimal path for 20 points

Another experiment with calculating the optimal path involves generating a path using the ant algorithm for 50 points. In this case, all turbines are situated in a random order, which rarely happens, but is suitable for testing the algorithm itself. The calculation result is shown in Figure 16.

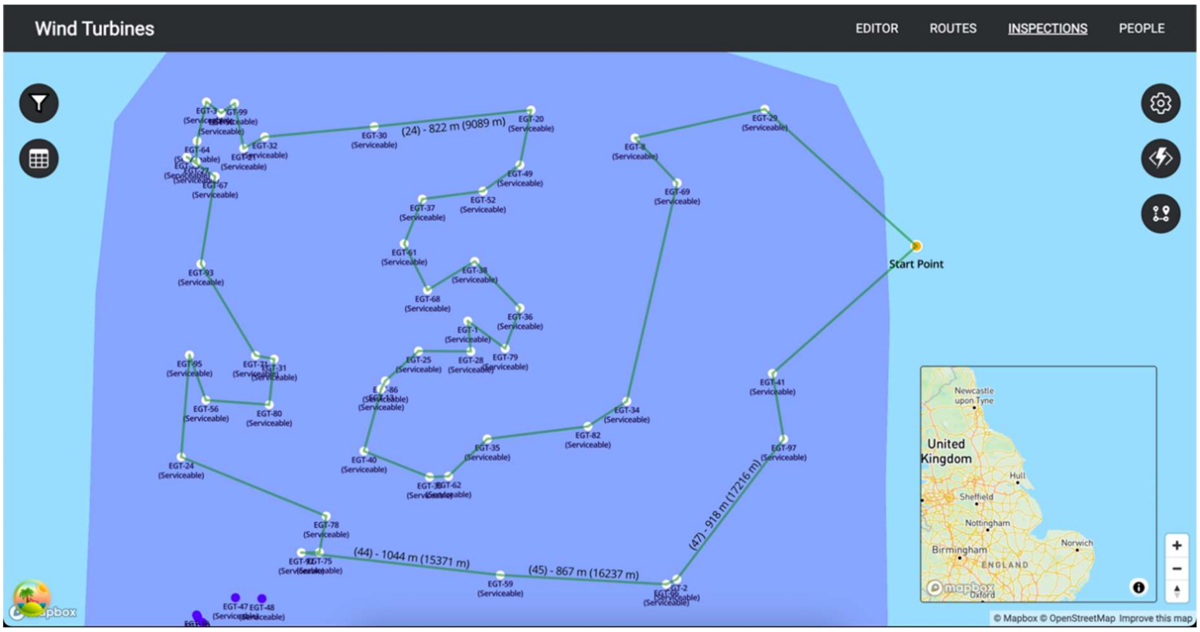

Figure 16: Result of calculating the optimal path for 50 points

The ant colony optimisation algorithm shows promising results in calculating the optimal path. However, it is not perfect either, as in some cases, it shows path generation artefacts, the so-called "loops", which can be seen in Figure 17.

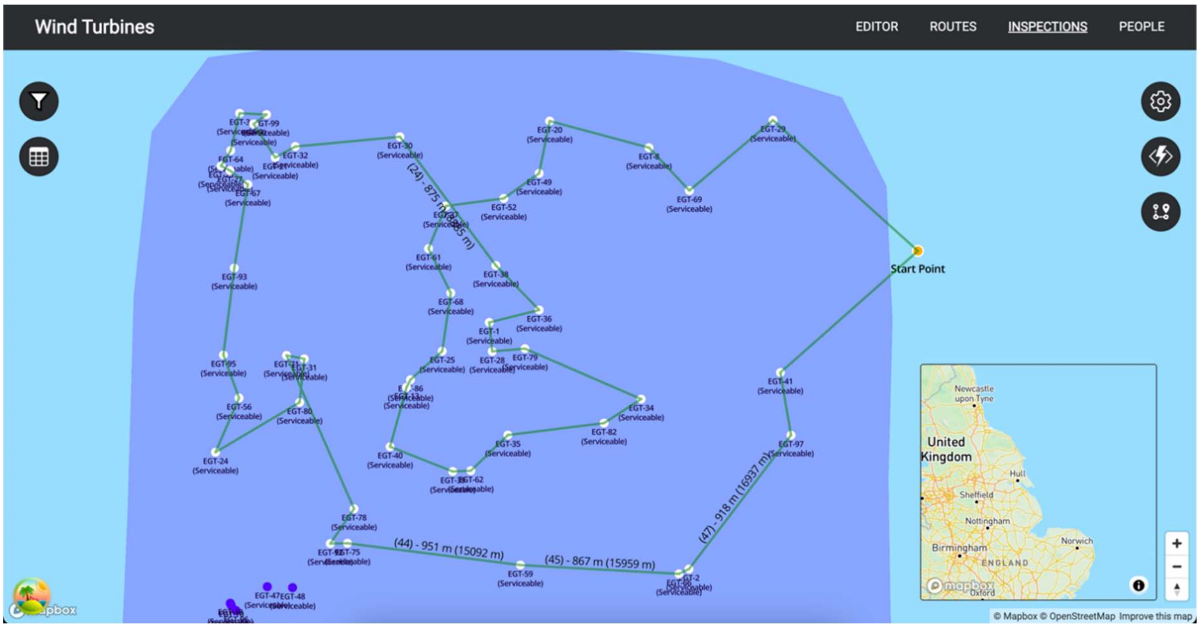

Figure 17: Formation of "loops" when calculating the optimal route

The main problem with these loops is the imperfection of the algorithm, which assumes that the segments of the path that the ants initially "walk" along will receive more weight and will be chosen as optimal. In contrast, if we compare these two previous paths in Table 3, we can see that due to the resulting "loop", the path has become longer.

Table 3
Comparison of paths without and with "loops"

#	Distance	Fuel spent
Without loops	17796 m	187.5 l
With loops	18285 m	191.8 l

Rules for finding intersections in the calculated route were developed. Since we can determine intersecting lines by direct enumeration and the mathematical operation of solving a system of equations in an acceptable short time, this method was chosen. After deciding the intersection location, the program takes all the points located between the intersecting sections of the route. It adds one more point at the beginning and end of the intersection for greater confidence in the calculations. Next, for all these points, the optimal path is calculated using the ant algorithm. An important point is a clear definition of the start and finish points of the route section, so as not to create new route artefacts. Since in this case there are not many points for calculation, the recalculation itself is performed in an acceptable amount of time. The option with the corrected route is shown in Figure 18. A comparison of distances and fuel consumption between the generated and corrected route options is shown in Table 4. Loop correction already gives a good result, reducing the total distance of the optimal path by 3%.

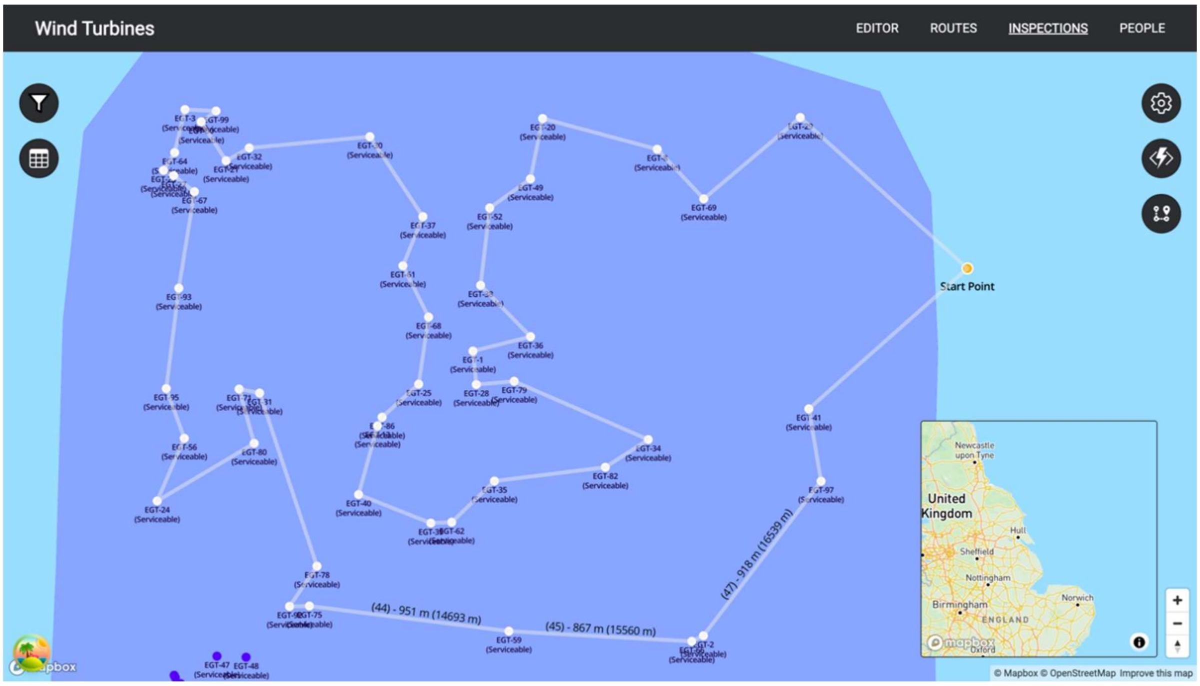

Figure 18: Corrected route

Table 4
Comparison of generated and corrected routes

#	Distance	Fuel spent
Generated	18285 m	191.8 l
Corrected	17886 m	188.1 l

The most interesting option for improving the route is to use an AI assistant with an LLM model that can independently determine intersections according to specified rules and improve the route. For this task, it is vital to train the model using queries that will tell the model which rules to use when searching for possible improvements to the path and how to calculate the route. The following query was used during training: "You are a model that can identify possible flaws of the optimal path generated by the ant colony optimisation algorithm. You will be provided with an array of points that conclude the path, and your task is to solve all issues you find. Each item of the array has a coordinates field with latitude and longitude. You need to find places where you see intersections and regenerate the area within them. You should keep the start and finish points in their places.

Additionally, analyse the path and try to optimise it as a whole thing." The AI assistant was able to solve the loop problem and transform the path itself. Still, the result was a less optimal path with a greater distance, so it was decided to remove the route improvement from the model training request and focus only on loop resolution. The comparison of the routes is shown in Table 5, and the visualisation is shown in Figure 19.

Table 4
Comparison of the path generated and the path corrected by the AI assistant

#	Distance	Fuel spent
Generated	18285 m	191.8 l
Corrected	20610 m	216.4 l

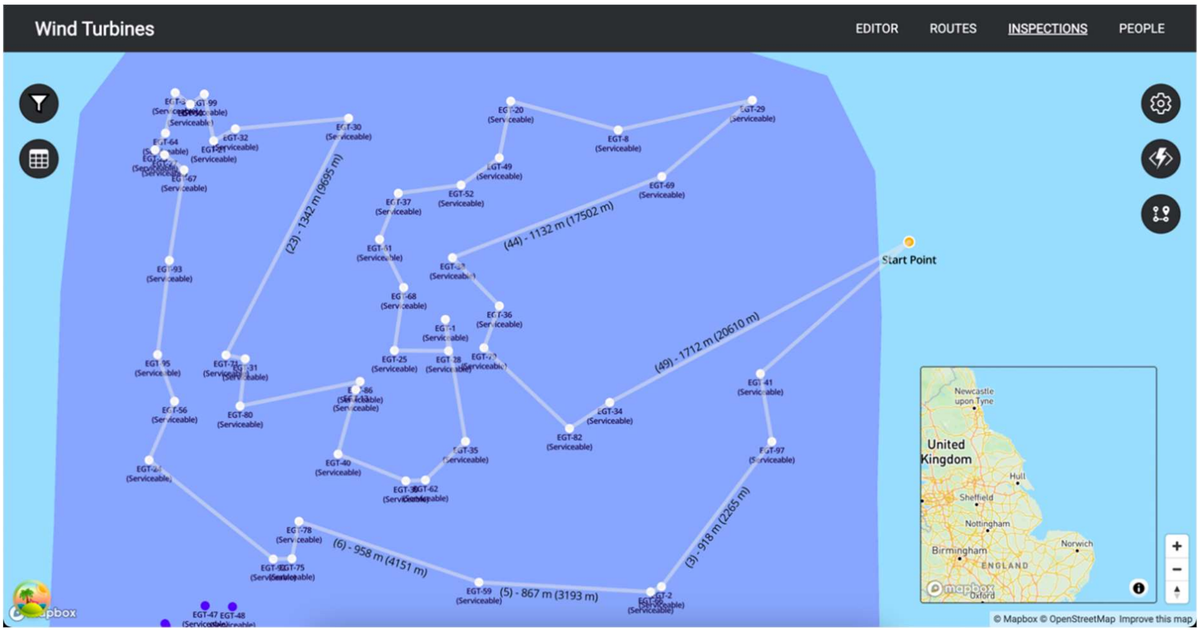

Figure 19: Result of route improvement using an AI assistant

5. Results

Experiments were conducted with the simulation of the real work of an operator or the system. First, several inspections were created to service turbines manually and with the help of an intelligent AI assistant. Using the AI assistant allowed for a significant reduction in the number of necessary actions, such as selecting turbines, adding commands, selecting dates and a ship to one message to the AI assistant, which then executes the command and transmits all the information as arguments. Experiments were also conducted using the AI assistant to set the starting point of the inspection and improve the result of calculating the optimal route using the intelligent assistant. The team of experts in the field confirmed the effectiveness of this solution.

Experiments with improving the result of the optimal route calculated using the ant colony optimisation algorithm yielded interesting results and motivation for further research.

The developed rules for finding intersections and re-programming the found sections using the ant algorithm allowed for effective improvement of the results of generating optimal routes, solving "loops" and bringing the result even closer to the optimal path. A series of experiments was conducted, the results of which are presented in Table 6.

Table 4
Comparison of the path generated and the path corrected by the AI assistant

#	Route type	Distance	Fuel spent
1	Generated	18285 m	191.8 l
	Corrected	18003 m	186.4 l
2	Generated	18663 m	201.8 l
	Corrected	18204 m	189.4 l
3	Generated	18499 m	196.7 l
	Corrected	17984 m	182.1 l
4	Generated	18142 m	190.9 l
	Corrected	18120 m	189.6 l
5	Generated	18918 m	212.5 l
	Corrected	18106 m	190.4 l

In general, using the software module to improve the results of optimal route generation, it is possible to achieve an average of 3-5% improvement in the optimal route in terms of distance compared to the initial generation using the ant algorithm.

On the other hand, completely transferring the process of improving the result of optimal route generation to the AI assistant and the LLM model does not always give the desired result in route optimisation. Very often, this leads not only to an extension of the route, but also to the formation of new "loops". Further experiments and training of the model using more data are required. Some experimental results are given in Table 7.

Table 4
Experiments with using an AI assistant

#	Route type	Distance	Fuel spent
1	Generated	18285 m	191.8 l
	Corrected	21305 m	236.2 l
2	Generated	18663 m	201.8 l
	Corrected	18503 m	196.8 l
3	Generated	18499 m	196.7 l
	Corrected	19403 m	215.1 l
4	Generated	18142 m	190.9 l
	Corrected	18985 m	215.8 l
5	Generated	18918 m	212.5 l
	Corrected	18505 m	196.7l

6. Discussions

The main objective of the work was to research the possibilities of using and implementing an AI assistant for software for the offshore turbine maintenance information system. Using AI allows reducing the time spent by system operators on using the product and performing daily actions, and provides assistance in intelligent technologies. The same assistant provides for the improvement of calculation results in optimal routes, which is the main functionality of the information system. Also, with the help of the assistant, it is possible to train future system operators.

The experiments conducted showed that the artificial intelligence assistant allows the operator to efficiently and quickly plan the maintenance of offshore wind turbines based on the rules of voice and text commands of the operator pre-set during training. At the same time, the assistant can determine the necessary command arguments and provide them along with the commands, providing possible services.

The module for improving the calculation results of the calculation path using the ant algorithm has also proven itself well. Among the improvements, we can highlight the developed rules for determining intersections - "loops" in the calculated route and local recalculation, which is based on the rules for generating and optimising the route by 3-5%.

An attempt to fully transfer the process of improving the finished result to the LLM side showed that further research and development of additional, more accurate tips are necessary, since in half of the cases, the path becomes less optimal.

The system is as close to reality as possible and provides fine-tuning of parameters, allowing the modelling of various situations for turbine maintenance planning and calculating optimal routes.

The results of the experiments showed that the software gives results close to optimal and significantly reduces the time for organising turbine maintenance compared to traditional interaction with the software. The software received high marks from industry experts.

Further development of the system includes improvement of the rules for determining intersections and artefacts of optimal route generation, as well as additional experiments with improving the finished result using LLM. Such an improvement will bring the ant colony optimisation algorithm to a new level, allowing for acceleration and increased generation accuracy. It will also enable calculations to be performed on a larger number of points. Everything depends on the demand for the technology in the subject area in which the information system operates. It includes deployment of the system in a production environment as an MVP to begin real testing by users who will use the software in their professional activities.

By collecting feedback on the system's operation and requests for additional functionality, it is possible to form a further list of functional elements that can be added to the decision support information system.

Now we can offer the following functionality for development:

- Development of a weather data collection subsystem;
- Adding recommendation features to the AI assistant;
- Enabling AI to independently collect commands for turbine maintenance.

The process of creating an inspection has already been described above; it consists of selecting the marine turbines that need maintenance, building teams for maintenance, setting maintenance dates and choosing a ship. Creating an inspection by the operator using the AI assistant reduces the number of actions required to write only one message.

It worth mentioning an idea for development, in close cooperation with specialists in the field of maritime transport, the implementation of support for streaming signals from sensors of real vessels that determine their position on the map, speed and course, their further processing, for example, displaying these parameters on the control equipment or creating records of turbine maintenance. It will allow for the collection of data for additional training of the AI model. The same can be solved for receiving data from turbines in real time on request and visualising the state of turbines on the user interface during their operation and maintenance.

7. Conclusions

The current state of the issue of intelligent information systems for automation of technological processes, including algorithms and technologies of work, functional capabilities, implementation of the interface and visualisation, and use of processing of text information using large text models, was analysed. On this basis, requirements were formed, and a module of the information system was developed to assist in decision-making for the organisation of maintenance of offshore wind power turbines.

The calculation of the optimal path using the ant colony optimisation algorithm and improvement of its result were also implemented.

The principles of using LLM models for processing text information were considered, and an intelligent assistant was developed, which allows for reducing the time required to perform operations in the system and helps in decision-making.

The principles of the ant algorithm, its description and formulas, and the possibilities of software implementation were also described. An improvement in the result of calculating the optimal path by recalculating sections of the route with intersections was proposed, which allows the path to be brought closer to a more optimal one.

The Experiment section described the developed software, which can be used to simulate the real work of an operator in his professional activity on organising the maintenance of turbines on offshore power plant farms. The provided architecture and description of the components give a brief explanation of the principles of the software and the technologies used. The experiments were described, where the main capabilities in working with the software of the system operator in the process of organising the maintenance of offshore turbines and calculating the optimal path were shown. The capabilities of working with an intelligent assistant were also demonstrated. The experiments were accompanied by a description of the results obtained when calculating the optimal path using the ant algorithm, as well as a comparison with the values obtained when improving the optimal path. A solution was developed for correcting "loops" after generating the optimal route. Rules for finding intersections of route sections and recalculating them were created. The measurements themselves were accurate enough to confirm the effectiveness of the methods for improving the optimal route, which allows improving the result by an average of 3-5%.

An experiment was also conducted using an intelligent assistant and an LLM model to improve the optimal route. Queries were developed for the model with rules on how to improve the optimal route. The experiments showed low efficiency of the solution now. The next step will be to continue improving the algorithm for calculating optimal routes and to better integrate the intelligent assistant into the system to simplify the operator's work and help in decision-making. It is also planned to continue experiments with improving optimal routes using an intelligent assistant and text-to-speech technologies as described in [30-31].

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly to check grammar and spelling. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Smelyakov, K., Chupryna, A., Darahan, D., Midina, S. Effectiveness of modern text recognition solutions and tools for common data sources, CEUR Workshop Proceedings, 2021, 2870, pp. 154–165, https://ceur-ws.org/Vol-2870/.

[2] Vysotska, V., Smelyakov, K., Sharonova, N., Derenskyi, M., Pliekhova, G., Repikhov, V. Information System for Monitoring and Planning Maintenance of Offshore Wind Farms, CEUR Workshop Proceedings, 2024, 3668, P. 63–82. Doi: 10.31110/COLINS/2024-2/006.

[3] Byzkrovnyi, O., Savulioniene, L., Smelyakov, K., Sakalys, P., Chupryna, A. Comparison of Potential Road Accident Detection Algorithms for Modern Machine Vision System, Vide. Tehnologija. Resursi - Environment, Technology, Resources, 2023, 3, pp. 50–55. doi: https://doi.org/10.17770/etr2023vol3.7299.

[4] Jiang, X., Lin, Z., He, T., Ma, X., Ma, S., Li, S. "Optimal Path Finding With Beetle Antennae Search Algorithm by Using Ant Colony Optimization Initialization and Different Searching Strategies," IEEE Access, vol. 8, pp. 15459-15471, 2020. Doi: 10.1109/ACCESS.2020.2965579.

[5] Owais, M., Alshehri, A. "Pareto Optimal Path Generation Algorithm in Stochastic Transportation Networks," IEEE Access, vol. 8, pp. 58970-58981, 2020. Doi: 10.1109/ACCESS.2020.2983047.

[6] W. Wang et al., "Development of an Adaptive User Support System Based on Multimodal Large Language Models," 2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Liverpool, United Kingdom, 2024, pp. 344-347, doi: 10.1109/VL/HCC60511.2024.00044.

[7] Niu, J., et al. "Design and Implementation of Intelligent Software Version Management Platform for Power Communication Network Based on Large Language Models," 2024 IEEE 4th International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 2024, pp. 1451-1555. Doi: 10.1109/ICIBA62489.2024.10868397.

[8] K. Qiu, S. Bakirtzis, I. Wassell, H. Song, J. Zhang and K. Wang, "Large Language Model-Based Wireless Network Design," in IEEE Wireless Communications Letters, vol. 13, no. 12, pp. 3340-3344, Dec. 2024, doi: 10.1109/LWC.2024.3462556.

[9] Krasnobayev, V., Kuznetsov, A., Yanko, A., Kuznetsova, T. "Solving the Shortest Path Problem Using Integer Residual Arithmetic," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 563-566. Doi: 10.1109/PICST51311.2020.9467947.

[10] A. M. Alshahrani, H. Farooq Ahmad and J. Hussain, "Detecting Health Misinformation on Social Networking Sites Using Large Language Models and Deep Learning-based Natural Language Processing," 2024 2nd International Conference on Foundation and Large Language Models (FLLM), Dubai, United Arab Emirates, 2024, pp. 561-568, doi: 10.1109/FLLM63129.2024.10852503.

[11] M. Li, "Exploring the Application of Large Language Models in Spoken Language Understanding Tasks," 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE), Jinzhou, China, 2024, pp. 1537-1542, doi: 10.1109/ICSECE61636.2024.10729345.

[12] C. Nicodeme, "Build confidence and acceptance of AI-based decision support systems - Explainable and liable AI," 2020 13th International Conference on Human System Interaction (HSI), Tokyo, Japan, 2020, pp. 20-23, doi: 10.1109/HSI49210.2020.9142668.

[13] Kyrychenko, I., Shyshlo, O., Shanidze, N. (2023) Minimising Security Risks and Improving System Reliability in Blockchain Applications: a Testing Method Analysis, CEUR-WS, 2023, v. 3403, Volume III: Intelligent Systems Workshop, pp. 423–433. ISSN 16130073.

[14] P. Papadopoulos, D. W. Coit and A. A. Ezzat, "Seizing Opportunity: Maintenance Optimisation in Offshore Wind Farms Considering Accessibility, Production, and Crew Dispatch," in IEEE Transactions on Sustainable Energy, vol. 13, no. 1, pp. 111-121, Jan. 2022, doi: 10.1109/TSTE.2021.3104982.

[15] H. Guo, "Research on Web Data Mining Based on Topic Crawler," in Journal of Web Engineering, vol. 20, no. 4, pp. 1193-1206, June 2021, doi: 10.13052/jwe1540-9589.20411. Abstract: This paper analyses the method of Web information data mining based on topic crawler.

[16] M. S, B. N M, S. N, M. H N, P. S and D. B L, "An Efficient Big Data Gathering in Wireless Sensor Network using Reconfigurable Node Distribution Algorithm," 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP), Bengaluru, India, 2022, pp. 1-6, doi: 10.1109/CCIP57447.2022.10058620.

[17] C. Minghua, L. Xiangqiang, Z. Fuxiao, Y. Dongchuan and L. Hua, "Application of remote sensing big data technology in refined urban management," 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 2020, pp. 397-400, doi: 10.1109/ICBAIE49996.2020.00089.

[18] O. M. Bakhriddin uali, A. J. Samadilla ugli and Z. S. G'olib ugli, "Microservice Approach In Designing A Scalable Architecture On The Example Of An Electronic Invoice System," 2021

International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2021, pp. 1-3, doi: 10.1109/ICISCT52966.2021.9670337.
[19] I. A. Kautsar, M. R. Maika, A. N. Budiman, A. B. Setyawan and J. Y. Awali, "Microservice Based Architecture: The Development of Rapid Prototyping Supportive Tools for Project Based Learning," 2023 IEEE World Engineering Education Conference (EDUNINE), Bogota, Colombia, 2023, pp. 1-6, doi: 10.1109/EDUNINE57531.2023.10102884.
[20] K. Adrio, C. N. Tanzil, M. C. Lianto and Z. E. Rasjid, "Comparative Analysis of Monolith, Microservice API Gateway and Microservice Federated Gateway on Web-based application using GraphQL API," 2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Palembang, Indonesia, 2023, pp. 654-660, doi: 10.1109/EECSI59885.2023.10295809.
[21] Y. Ling, C. Yang, X. Li, D. Bin, S. Han and M. Xie, "WEB Security Protection Technology Based on Honeypot Technology," 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), Dalian, China, 2022, pp. 112-115, doi: 10.1109/TOCS56154.2022.10016100.
[22] Kyrychenko, I., Pronina, D., "Comparison of Redux and React Hooks Methods in Terms of Performance", 2022 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022), 2022. – CEUR-WS 3171, 2022, ISSN 16130073. - Volume I: Main, PP. 791 - 800.
[23] C. Curto, D. Giordano, D. G. Indelicato and V. Patatu, "Can a Llama Be a Watchdog? Exploring Llama 3 and Code Llama for Static Application Security Testing," 2024 IEEE International Conference on Cyber Security and Resilience (CSR), London, United Kingdom, 2024, pp. 395-400, doi: 10.1109/CSR61664.2024.10679444.
[24] W. Wang, J. Guo, Z. Li and R. Zhao, "Behavior model construction for client side of modern web applications," in Tsinghua Science and Technology, vol. 26, no. 1, pp. 112-134, Feb. 2021, doi: 10.26599/TST.2019.9010043.
[25] Q. Wan and S. Zhang, "Online Teaching System of Advertising Art Design Course Based on Redux with Secure Shell Framework," 2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 2024, pp. 1-5, doi: 10.1109/ICIICS63763.2024.10859841.
[26] V. Tagdiwala, A. Bharoliya, P. Patel, D. Shah and M. Aibin, "Robust Client and Server State Synchronisation Framework For React Applications: react-state-sync," 2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Regina, SK, Canada, 2023, pp. 475-481, doi: 10.1109/CCECE58730.2023.10289106.
[27] M. Kim, J. Choi, D. Kim and Y. M. Ro, "Textless Unit-to-Unit Training for Many-to-Many Multilingual Speech-to-Speech Translation," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, pp. 3934-3946, 2024, doi: 10.1109/TASLP.2024.3444470.
[28] V. M. Reddy, T. Vaishnavi and K. P. Kumar, "Speech-to-Text and Text-to-Speech Recognition Using Deep Learning," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 657-666, doi: 10.1109/ICECAA58104.2023.10212222.
[29] S. Liu, A. S. Hussain, C. Sun and Y. Shan, "Music Understanding LLaMA: Advancing Text-to-Music Generation with Question Answering and Captioning," ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, pp. 286-290, doi: 10.1109/ICASSP48485.2024.10447027.
[30] V. Kovtun, V. Vysotska, Oksana Kovtun, Contextual-formal concept of creating explainable models for corpus linguistics tasks, Ceur Workshop Proceedings, Vol-2005, 109-126, https://ceur-ws.org/Vol-4005/paper9.pdf
[31] M. G. Gonzales, P. Corcoran, N. Harte and M. Schukat, "Joint Speech-Text Embeddings for Multitask Speech Processing," in *IEEE Access*, vol. 12, pp. 145955-145967, 2024, doi: 10.1109/ACCESS.2024.3473743.