



# Multi-agent system architecture for winter road maintenance: a real Spanish case study

Diego M. Jiménez-Bravo<sup>1,2</sup> · Javier Bajo<sup>2</sup> · Jacinto González-Pachón<sup>2</sup> · Juan F. De Paz<sup>1</sup>

Received: 2 June 2022 / Revised: 15 April 2024 / Accepted: 23 April 2024 /

Published online: 18 May 2024

© The Author(s) 2024

## Abstract

Road safety remains a critical issue in contemporary society, where the sudden deterioration of road conditions due to weather-related natural phenomena poses significant risks. These abrupt changes can lead to severe safety hazards on the roads, making real-time monitoring and control essential for maintaining road safety. In this context, technological advancements, especially in sensor networks and intelligent systems, play a fundamental role in efficiently managing these challenges. This study introduces an innovative approach that leverages a sophisticated sensor platform coupled with a multi-agent system. This integration facilitates the collection, processing, and analysis of data to preemptively determine the appropriate chemical treatments for roads during severe winter conditions. By employing advanced data analysis and machine learning techniques within a multi-agent framework, the system can predict and respond to adverse weather effects swiftly and with a high degree of accuracy. The proposed system has undergone rigorous testing in a real-world environment, which has verified its operational effectiveness. The results from the deployment of the multi-agent architecture and its predictive capabilities are encouraging, suggesting that this approach could significantly enhance road safety in extreme weather conditions. Furthermore, the proposed architecture allows the system to evolve and scale over time. This paper details the design and implementation of the system, discusses the results of its field tests, and explores potential improvements.

Diego M. Jiménez-Bravo, Javier Bajo, Jacinto González-Pachón and Juan F. De Paz contributed equally to this work.

✉ Diego M. Jiménez-Bravo  
dmjimenez@usal.es, djimenez@fi.upm.es

Javier Bajo  
jbajo@fi.upm.es

Jacinto González-Pachón  
jgpachon@fi.upm.es

Juan F. De Paz  
fcofds@usal.es

<sup>1</sup> Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain

<sup>2</sup> Departamento de Inteligencia Artificial, ETSI Informáticos, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain

**Keywords** Embedded agents · Machine learning · Multi-agent systems · Parallel behavior · Road maintenance · Road monitoring

## 1 Introduction

Road safety is a fundamental factor in twenty-first-century society. With globalization, people are more and better connected through networks of roads, flights, trains, etc. In this sense, safety at both transport and road level is essential. This idea has also reached the highest European authorities, where the European Parliament has released a report containing a series of guidelines to be followed reflected in the EU Road Safety Policy Framework 2021–2030 [1]. In this report, the European leaders have defined certain recommendations related, for example, to road safety.

In this sense, road safety involves both the maintenance of the road itself due to deterioration and/or misuse of the road or, among other things, due to weather conditions. Weather conditions have a great impact on road safety and can be one of the main causes of accidents [2]. Consequently, it is necessary to provide good road maintenance in adverse weather conditions.

Therefore, to be able to offer citizens good traffic conditions even when the circumstances are adverse, it is necessary first of all to monitor the state of the roads in real time. Hence, it is essential to deploy stations or sensor networks along the roads to store and study the road conditions. Communication technologies play a key role in this task. It is not only necessary to monitor the state, but also to develop intelligent systems based on artificial intelligence (AI) that can discover, identify, and even act in the face of events or circumstances that occur on the roads and that could put the safety of road users at risk.

On the other hand, such systems not only improve safety but also allow road maintenance companies to automate the monitoring process and reduce costs in terms of fuel or chemical components.

In this sense, this article proposes a system based on a multi-agent architecture capable of monitoring and predicting when it would be necessary to apply a certain chemical treatment on road sections in adverse winter conditions.

### 1.1 Background

#### 1.1.1 Road monitorization and maintenance

Roads are connections between population areas that allow citizens to travel between areas, using in many cases motorized vehicles. Hence, the proper state of the roads is crucial to prevent possible traffic accidents or dangerous situations. In this sense, the scientific community has done several pieces of research focused on the monitoring and maintenance of roads.

When it comes the time to analyze those studies, we are going to separate the monitorization research items from the maintenance researcher items.

*Road monitoring* Road monitoring consists of the measure of relevant features related to the state and conditions of the roads. These measurements could be conducted for several reasons, for instance, the monitorization of roads for avoiding or identifying structural damages in the pavements. Some recent studies for this line of research are [3, 4]; the articles are focused on the use of accelerometers and gyroscopes to detect and even classify road anomalies.

However, although road anomalies are important, this study is more focused on the monitoring of road conditions such as temperature, humidity, and weather conditions. The analysis of this type of variable is crucial as well for road maintenance, especially when there are extreme weather conditions such as winter conditions.

For instance, Chapman and Bell [5] have deployed sensor and humidity sensors based on LPWANs networks for road monitorization in the UK. This study establishes an approach for monitoring conditions with the help of sensor networks. In the same way, [6] illustrates the use of other more complex fixed sensor networks used for air temperature monitoring, road surface monitoring, dew point measuring, wind speed monitoring, etc. Nonetheless, [6] also indicates that mobile sensors located in a car can help to control this types of variables.

Furthermore, other approaches can be used for monitoring the surface conditions of the road. Pan et al. [7] make use of cameras to infer the state of the road in winter conditions. However, this input seems to be less rich than the one obtained with sensor networks, nonetheless, the image can help to gain visibility about the state of the road apart from the values measured with sensors on the road.

Taking this into account, it seems that the ideal road surface monitoring for winter conditions could be a combination of several sensors monitoring variables related to the surface and environment conditions and the use of a camera for a better understanding of these measures.

*Road maintenance* Road monitoring seems crucial to performing another important task, road maintenance. Road maintenance is based on the data provided by road monitoring; therefore, these two tasks form a pipeline process. Also, thanks to the data obtained in the monitorization it is possible to automate road maintenance. Particularly, the winter road maintenance allows companies and governments to apply the right chemical treatment to each section of the road in time.

Hence, several researchers have worked in this line of research. For instance, Hatamzad et al. [8] developed a neuro-fuzzy system trained with real data to predict the road surface friction coefficient. According to the authors, the system obtains an excellent accurate prediction. The system may help decision-makers to apply a chemical treatment on roads in advance.

Similarly, Pan et al. [7] have developed various systems based on CNN (convolutional neural network) architectures to detect the level of snow in road images. In this case, the system can be defined as a decision helper for road maintainers and even citizens. The scientific article concludes that the results are encouraging.

Another interesting approach is the one illustrated by Matthews et al. [9]; they refine the Winter Severity Index using public data and apply it to evaluate its significance over winter weather trends. With that information, they were able to model the impact of weather on winter road maintenance in advance. This model can be helpful in the same ways as the previous ones were, as a decision helper.

Nonetheless, road maintenance is not only based on winter conditions. Furthermore, studies like [10] applied computer vision techniques to detect damage to road surfaces, road markings, and traffic signs. This information is also important for the safety of the users. Therefore, other articles have been published in recent years, such as [11–13].

### 1.1.2 Multi-agent systems

Multi-agent systems (MASs) can be defined as a combination of interacting agents that share resources and establish communications between them to accomplish their goals and the global goal of the system. These types of systems have been applied to different domains such as IoT (Internet of Things) [14–16], medicine [17, 18], among others.

However, according to the best of our knowledge, there have not been previous implementations of multi-agent systems in the area of road winter maintenance. Nonetheless, MASs have been applied for several tasks related to the roads. For instance, [19] proposed a traffic light control based on a MAS. More specifically, they use a pheromone-based hierarchical MAS.

Another recent study developed in a road environment is the one proposed by Małecki [20]. He proposed a traffic flow simulation for street parking based on a MAS. Another important area where MASs are applied is in the area of autonomous driving. As an example, [21] mixes reinforcement learning and MAS to research autonomous driving. An interesting area in which MASs have a tremendous impact is reinforcement learning.

*Intelligent agents* An interesting aspect of MASs is the definition of intelligent agents: agents that can extract knowledge from data and use it to accomplish their goal and/or the goal of the MAS. These types of agents are usually used in IoT environments. A good example is [22], which uses a MAS based on intelligent agents for data aggregation in wireless sensor networks. In the same way, Kottenko et al. [23] used intelligent agents for anomaly detection in an IoT network.

*Embedded agents* An important issue when someone implements a MAS in an IoT environment is how to include IoT devices into the multi-agent platform. Therefore, researchers have incorporated the term embedded agent into this complex ecosystem. These types of agents are the software agents included in IoT devices that have direct communication with the MAS. Thus, it is important to establish the connection between the physical IoT object and the software multi-agent platform. According to the authors of [24], this connection is done via a middleware; hence, the connection can be established under different protocols depending on the conditions of the IoT devices and the requirements of the IoT and MAS platform.

Different authors have included embedded agents in their studies, for instance, the already cited article developed by De La Iglesia et al. [24] explains the configuration of a bicycle IoT ecosystem along with a MAS using embedded agents. Villarrubia et al. [25] also developed a MAS integrated with IoT devices for home emergencies. Also, the studies [26, 27] included embedded agents in physical devices. Nonetheless, none of these articles have studied or implemented the use of embedded agents for winter road maintenance.

## 2 Proposed system

As has been mentioned previously, the proposed system is based on a MAS architecture. The main goal of this system is to establish the chemical procedure to perform over roads in the winter months. The proposed system is illustrated in Fig. 1. With the aim of better explaining the proposed system, we will explain the different parts of it in different subsections.

### 2.1 App layer

The app layer is one of the two layers of the proposed MAS. It can be defined as the main layer of the proposed system since it performs road maintenance and monitoring tasks. These tasks are performed by its different agents, and these are explained below:

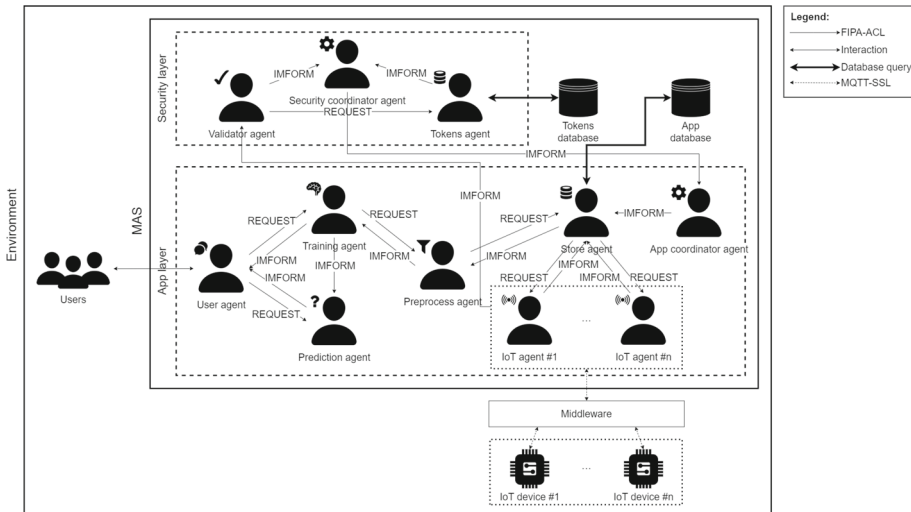


Fig. 1 Multi-agent system proposed

### 2.1.1 Sensor agents

Sensor agents can be defined as embedded agents connected with physical devices with an Internet connection. (In this case, Internet connection means that the device implements the OSI model and its network layer implements the TCP/IP protocol.) So, the connection between the software agents and the physical devices is done via middleware; this strategy is crucial so the embedded agent can run independently from the hardware devices [24].

Hence, the physical IoT devices send the measurements to the middleware of the proposed system. This middleware is responsible for sending the data to the specific sensor agent. In the case of the presented system, we use the MQTT (Message Queuing Telemetry Transport) protocol over SSL (Secure Sockets Layer). This protocol specifically designed for IoT platforms allows encrypted and light communications between the IoT devices and the MAS via a Publish/Subscribe message queue.

Bearing that in mind, the different sensor agents configured in the MAS implement different types of behaviors:

- **Sensor listening cyclic behavior:** is a cyclic behavior that listening to the MQTT message received from the middleware. When the sensor agent receives measurements it stores them in a temporal buffer. However, not all the data received by the sensor agents are trusted. Therefore, this behavior also informs the security layer (see 2.2) of the MAS to check if the IoT device is trusted.
- **Store data cyclic behavior:** this cyclic behavior listens for a “Request” message from the store agent. At this time, it is understood that the data stored in the temporal buffer is trusted, so, the sensor agent sends this data to the store agent and freezes the buffer.

### 2.1.2 App coordinator agent

The app coordinator agent is the coordinator of the app layer. It has communication with the other layer of the MAS (security layer). Thus, the security layer informs this agent if the

IoT is trusted; if so, the app coordinator informs the store agent that the received data can be stored in the app database. The behavior of this agent is explained below:

- **App coordinator cyclic behavior:** this cyclic behavior waits for an “Inform” message coming from the security layer and if received it informs (“Inform” message) the store agent that the IoT device is trusted and therefore its data can be stored in the MAS database.

### 2.1.3 Store agent

The store agent is responsible to get and collect the data in the app database. The behaviors of this agent are the following:

- **Storing cyclic behavior:** this behavior waits for an “Inform” message from the app coordinator agent; once received this behavior sends a “Request” message to the different sensor agents and stores the data received from them via an “Inform” message in the app database.
- **Obtain data cyclic behavior:** this cyclic behavior waits for a “Request” message from the preprocess agent to obtain the requesting data from the app database. Also, the behavior sends an “Inform” to the requesting agent with the ordered data.

### 2.1.4 Preprocess agent

The preprocess agent is responsible for cleaning and preprocessing the data correctly so it is in the right and expected format to perform the training task correctly. The behavior performed by this agent is as follows:

- **Preprocessing cyclic behavior:** this cyclic behavior waits for a request message coming from the train agent. As explained, its main functionality is to preprocess the data; but first, it has to “Request” and obtain the data from the store agent. After that, the behavior can perform its main task. In the specific case of this article, the preprocessing tasks consist of removing instances with missing values, obtaining the instance with the minimum temperature for every day in the data, and applying a correlation attribute evaluation to the selected instances. Thus, after applying the preprocessing tasks, the behavior obtains the dataset and it sends an “Inform” message about it to the train agent.

### 2.1.5 Train agent

Train agent deals with the training of different machine learning (ML) algorithms using a dataset obtained from the sensorized data stored in the app database. The exact behavior of this agent is explained in the following paragraph.

- **Train models cyclic parallel behavior:** this behavior trains different machine learning algorithms with the same dataset configuration to compare each of them and select the best one according to a metric value (accuracy). However, the internal behavior is quite specific; since the agent can train several behaviors (the exact number depends on the user preferences), it is crucial to optimize its implementation. Therefore, each model is trained parallel in a thread behavior; this strategy allows for optimizing the system in terms of time. To a better understanding of this behavior, a schema of it is shown in Fig. 2. Once the training is performed, the behavior sends an “Inform” the user agent of the results and informs the prediction agent of the best model.

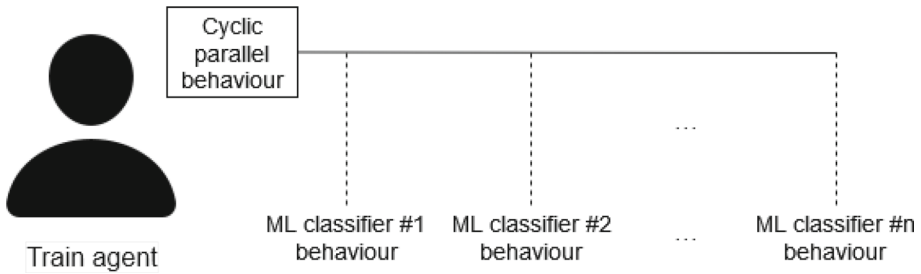


Fig. 2 Train models cyclic behavior

### 2.1.6 Predict agent

The prediction agents address the task of predicting the road maintenance chemical process. It receives an unlabeled instance and returns the chemical process using the best-trained model. The definition of the behavior is the following:

- **Model importing cyclic behavior:** it is a cyclic behavior that waits for an “Inform” message from the train agent. This message contains the best-trained model that will be used for prediction purposes.
- **Prediction cyclic behavior:** it is a cyclic behavior that waits for a request of the user agent and performs the prediction using the model received in the model importing cyclic behavior. After it performs its task from the user agent with the prediction result.

### 2.1.7 User agent

The user agent is responsible for interacting with the user itself. That is to say, it is the agent that interacts with the environment. The user agent has two different behaviors:

- **Train cyclic behavior:** the first behavior of the user agent is to get in contact with the users of the system. The user is responsible for feeding the user agent with ML algorithm/s to train. Hence, once the agent has the ML algorithm/s, it sends a “Request” message to the train agent.
- **Display results cyclic behavior:** this behavior is blocked until the agent receives an “Inform” message from the train agent. Its labor consists of displaying the results provided by the train agent or the prediction agent in its message. With this interaction, the user can understand the performance of the different ML algorithms selected by the user in the previous behavior or the prediction of an instance.
- **Predict cyclic behavior:** it is responsible for sending a “Request” message to the prediction agent to obtain the label for an unlabeled instance.

## 2.2 Security layer

The security layer is the second layer of the proposed MAS. As it may be interpreted by its name, its functionality is related to the security and confidence of IoT devices of the proposed system. In the next subsections, we will explain its different agents and their behaviors.

### 2.2.1 Validation agent

The validation agent acts as an intermediate agent before the validation takes place. It receives an “Inform” message every time the device’s agents receive data. It is important to validate the IoT device every time the MAS receives new data to maintain the integrity and internal state of both, the MAS and the app database. Its behavior is:

- **Validation cyclic behavior:** this cyclic behavior receives an “Inform” message with the token related to the IoT devices. Its functionality is to request the validation of this taken to the tokens agent via a “Request” message.

### 2.2.2 Tokens agent

The tokens agent is in charge of validating or not the tokens received from the IoT devices. To accomplish its task it validates the tokens against a tokens database in which all trusted devices’ tokens are stored. Below, we explain its behavior.

- **Tokens validation cyclic behavior:** it receives via a “Request” message the token/s to validate. As explained, the agent consults if the token/s is stored in the tokens database. After this evaluation, the tokens agent sends an “Inform” message to the security coordinator agent.

### 2.2.3 Security coordination agent

This agent is the coordinator of the security layer. It is in charge of the communication between the two layers of the system after the token validation process takes place. It is a single cyclic behavior:

- **Validation response cyclic behavior:** it sends an “Inform” message to the app agent if the IoT devices from which the system has received data are trusted or not. This message is based on the content of the also “Inform” message received from the tokens agent.

## 3 Case study

The previously presented system is deployed in a real environment. Hence, we can evaluate its performance in a real situation. This section explains different aspects concerning the case study of the proposed system.

### 3.1 Sensorized environment

One of the key aspects of the proposed system is the sensorization of the environment. The data will provide our system with knowledge and decision-making power. In this sense, we have five sensorized systems on the A-62 highway between Salamanca (Spain) and Vilar Formoso (Portugal). These sensor systems monitor different variables and give us the context of their situation and environment. A schematic view of the monitorization system is illustrated in Fig. 3.

These sensing platforms inform about their geospatial location, altitude, and visual information about the state and traffic on the road. As we mentioned in 1.1, the fact that the system

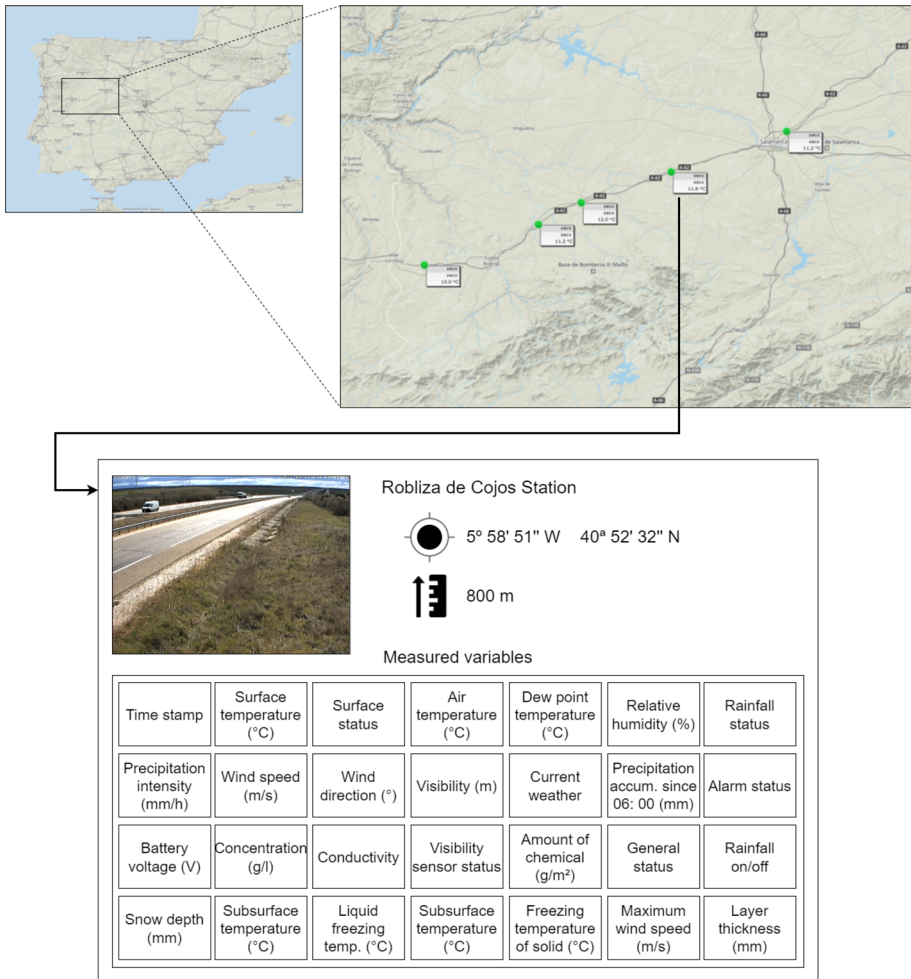


Fig. 3 Case study outline

incorporates visual data for the decision-making process is important to give the users a better context. However, in the presented system, this visual data is not used for the training process or the prediction process; these two steps are performed only with non-visual data measured with the sensors placed in the sensorized systems. The variables measured are listed below along with their units:

- Surface temperature (°C).
- Surface condition.
- Air temperature (°C).
- Dew point temperature (°C).
- Relative humidity (%).
- Rainfall status.
- Precipitation intensity (mm/h).
- Wind speed (m/s).

- Wind direction (°).
- Visibility (m).
- Current time.
- Accumulated precipitation since 06:00 local time (mm).
- Alarm status.
- Battery voltage (V).
- Concentration (g/l).
- Conductivity.
- Visibility sensor status.
- Amount of chemical (g/m<sup>2</sup>).
- General status.
- Rain on/off.
- Snow depth (mm).
- Subsurface temperature (typically at a depth of 30 cm/12 inches) (°C).
- Liquid freezing temperature (°C).
- Subsurface temperature (depth of 6 cm/2.4 inches) (°C).
- Freezing temperature of the solid (°C).
- Maximum wind speed (m/s).
- Layer thickness: water equivalent (mm).

In the current study, we will implement the proposed system with only one of the five stations. The station used for the case study is the station located close to Robliza de Cojos (Spain) (5°58'51"W, 40°52'32"N). This station and its sensorized system are connected via the MQTT protocol with the MAS proposed. The connection is performed as explained in 2.1.

### 3.2 Dataset

As mentioned above, this case study will focus on one of the monitored stations. Therefore, the dataset used for training the models in the MAS will only contain data from Robliza de Cojos' station.

The dataset contains a total of 84,787 instances. These instances are comprehended between 11/01/2019 and 30/01/2022. Also, the sensorized platform measured data every 10 minutes; hence, the system obtained 144 records per day. However, the system is not active during the whole year; it only collects environment and road data from November to May; this period corresponds to the most extreme winter months in the area of the sensorized platform. Moreover, every instance of the dataset contains a value for each of the variables listed in the previous subsection.

Nonetheless, and following the indications of the road maintenance experts, not all the records and variables are useful for decision support. The experts only focus on the "Surface temperature (°C)" variable to determine the road's chemical treatment. They used the minimum "Surface temperature (°C)" to determine the action to perform. Taking that into account, we filter our dataset by selecting the record with the minimum "Surface temperature (°C)" per day; however, unlike road maintenance experts, we do take into account the rest of the variables for treatment prediction. As a consequence, the filtered dataset reduces its number of instances to 447.

Between these instances there are different classes, "no\_treatment," "humidified\_preventive," "humidified\_curative," "curative\_salt," "preventive\_salt," "preventive\_brine." Hence, the distribution of these classes in the dataset is detailed in Table 1.

**Table 1** Dataset classes distribution

Class label	Number of instances
No_treatment	330
Humidified_preventive	28
Humidified_curative	1
Curative_salt	1
Preventive_salt	6
Preventive_brine	81

### 3.3 Multi-agent system implementation

The system proposed in 2 has to be developed to test its performance in a real situation environment. So, the MAS is implemented using the JADE multi-agent platform [28]. JADE (Java Agent Development Framework) is a multi-agent framework developed in the Java programming language. There are other multi-agent frameworks such as JADEX [29], PADE [30], SPADE [31], and osBrain [32] among others. Nevertheless, JADE was selected because of its extended use in the MAS community and the facility that it gives to design cyclic parallel behaviors.

Thus, the MAS is implemented within the JADE framework following the design illustrated and explained in 2. However, it is important to mention the ML algorithms included in the Train agent. Hence, for this case study, this agent will contain several ML algorithms listed and explained briefly below:

- C4.5: it is an ML algorithm based on a decision tree. It was proposed by Quinlan [33]. To obtain the output the algorithm builds a decision tree while training following two main principles, gain function, and gain rate function.
- Random Forest: a classification ML algorithm proposed by Breiman [34]; as may be interpreted by its name, it uses several random trees. These algorithms are conformed using the Bagging algorithm and the bootstrapping techniques. In this case, each random tree of the random forest algorithm performs classification and the majority vote policy is applied to obtain the output of the algorithm.
- RIPPER: The Repeated Incremental Pruning Produce Error Reduction (RIPPER) is a rule-based ML algorithm [35]. It has two fundamental stages, a rule creation stage where multiple rules are obtained until 100% accuracy is achieved, and a pruning stage. These stages are repeated iteratively until several conditions are met. In addition, the RIPPER algorithm applies these two stages to each of the resulting rules.
- $k$ -Nearest Neighbors ( $k$ -NN): the  $k$ -NN algorithm is an instance-based algorithm proposed by Aha et al. [36]. It looks for similarities between different instances in the training set to determine the class of a new instance; in other words, it looks at the closest instances.
- Support Vector Machine (SVM): this type of algorithm makes use of support vectors to establish the areas corresponding to each of the classes present in the training set [37]. These vectors are constructed only by taking into account the instances present in the peripheries of the  $n$ -dimensional set formed by the different classes.
- Naive Bayes: it is an ML algorithm that tries to classify an instance using the Naive Bayes theorem and probabilities [38].

The purpose of implementing several ML algorithms is to test the cyclic parallel behavior under controlled circumstances. The main idea behind this specific type of behavior is to train several classifiers at the same time and select the best one for the MAS. It is important to mention that the training strategy followed in the training process is the tenfold cross-validation. Nonetheless, to clarify which of the ML algorithms is the best, it is crucial to define the metrics used in the case study.

### 3.4 Evaluation metrics

To evaluate the different ML algorithms, we will use several metrics. It is important to mention that the problem that the different ML algorithm aims to resolve is a classification problem. Therefore, the metrics used in this case study will focus on that task.

The used metrics are listed below:

- Accuracy: it determines the percentage of corrected classified instances. It is calculated as shown in Eq. 1,

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

where TP (true positives) is the number of positive instances classified as positive, TN (true negatives) is the number of negative instances classified as negative, FP (false positives) is the total number of instances wrongly classified as positive instances, and FN (false negatives) is the total number of positive instances classified as negatives.

- The area under the curve: the area under the curve (AUC) of a classifier is equal to the probability that the classifier tags a randomly chosen positive instance higher than a randomly chosen negative instance. As higher the AUC, better the classifier. The AUC is calculated as the area under the curve formed with the false positive rate (FPR) and the true positive rate (TPR).
- Kappa statistic: the Kappa metric or Cohen's Kappa ( $k$ ) is a value that measures the level of agreement between two predictors. In this sense, it tries to measure the percentage of agreement based on the concordance that could be expected due to pure chance. The higher the percentage, the higher the agreement and therefore the higher the classifier. Its equation is shown in Eq. 2,

$$k = \frac{p_o - p_e}{1 - p_e} \quad (2)$$

where  $p_o$  is the relative observed agreement among raters/predictors, and  $p_e$  is the hypothetical probability of chance agreement.

- Mean Absolute Error (MAE): it is a metric that measures the absolute error in each of the predictions by subtracting the predicted value from the actual value adding the absolute value of each of these operations and then dividing it by the total number of predictions. The smaller this value is, the better the classifier. The equation is shown in Eq. 3,

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (3)$$

where  $n$  is the total number of predictions,  $y_i$  is the prediction of a instance, and  $x_i$  is the actual class of the instance predicted.

## 4 Results

An important aspect of every research is to test the proposed system according to the specifications defined in the case study planned. Therefore, this section shows the results obtained by the proposed system; we will divide this section into several subsections.

### 4.1 Multi-agent system deployment

The main contribution of this research article is the MAS for winter road maintenance assistants. As explained, the different agents and entities collaborate between them to achieve a common goal. In this sense, the proposed architecture has shown a good performance related to agents' communications and the effectiveness of every single agent in the system. It is crucial to a good performance in terms of communications since it allows a good execution of the whole system. These communications are done quickly and without significant delays. Hence, after observing the behavior of the different agents within the system, it can be said that the system works perfectly for this kind of system.

### 4.2 Train agent results

Without any doubt, another key contribution of this work is the cyclic parallel behavior that we have defined in the train agent. In particular, this specific behavior allows the MAS to train different ML algorithms at the same time. In Sect. 3.3, we define the different ML algorithms that will be trained to test the system and validate the prediction system using a tenfold cross-validation strategy over the presented dataset (see 3.2).

In this preliminary solution, we show the results of six different ML models. These results are illustrated in Table 2. We can also see the ROC curves illustrated in Fig. 4 of the two best models, random forest and C4.5.

**Table 2** Machine learning algorithms results

ML algorithm	Accuracy	AUC	Kappa statistic	MAE
C4.5	0.8210	0.8550	0.5542	0.0634
Random Forest	0.8501	0.9619	0.5845	0.0887
RIPPER	0.8188	0.8417	0.5364	0.0798
<i>k</i> -NN	0.7875	0.8166	0.4887	0.0739
SVM	0.8389	0.8541	0.5900	0.2277
Naive Bayes	0.7584	0.9362	0.4961	0.0808

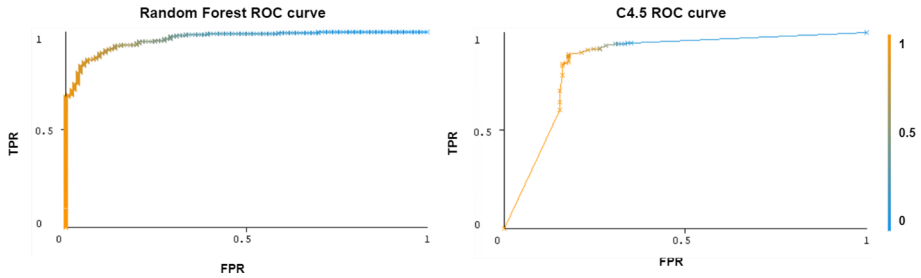


Fig. 4 ML algorithms' ROC curves

## 5 Discussion

In this section, we will discuss the results obtained in this research. We will divide this section into two subsections.

### 5.1 Multi-agent system deployment

As mentioned above, the multi-agent platform has performed satisfactorily and its distribution and composition have proven to be effective for the task of monitoring and predicting winter road treatments.

The choice of an IoT system based on the multi-agent architecture proposed in 2 is based on the advantages provided by multi-agent systems. Among these advantages, we can highlight the scalability of the system in case it is necessary to add new IoT devices and therefore the consequent sensor agents. The inclusion of agents that manage IoT devices allows these agents to specialize in the different geographical areas where the sensors are located. In addition, MASs are highly adaptable to rapid changes in the environment, such as in this specific case climate changes, allowing for an effective and timely response. Finally, the ability to process data at the edge of the network facilitates real-time decision-making without relying on a centralized data center, improving the speed and efficiency of response to critical conditions. These advantages not only allow the system to work properly in its current state but also allow the system to keep evolving and adapting to changes that might emerge in the future.

The only downside of the proposed system is the delay and message loss caused on some occasions by the MQTT protocol that provides the MAS with the sensors' data. However, this problem is not as important as it may sound; because as explained in the previous section the training process is done with the minimum "Surface temperature (°C)" record per day, and the prediction is not performed for every collected record. Consequently, this downside is not relevant to our proposed system.

### 5.2 Train agent results

According to the results, we can see that the random forest algorithm obtains the best metrics after the cross-validation. In general terms, we can see that the algorithms based on rules and/or trees along with the SVM algorithm are the ones that obtain a better performance in the training process. Hence, the random forest model is going to be predefined in the MAS in case the user declines the option to train a new model with new data.

Moreover, the behavior of the train agent has been demonstrated to be very effective in obtaining several ML models in a single process. This feature is crucial since it is one of the key values of the proposed system.

However, looking also at the results, it can be seen that there is still a significant improvement margin as far as the metrics are concerned. It is especially significant to observe the kappa value where the margin for improvement is enormous and the results obtained may leave some doubts about the effectiveness of the classifiers. Moreover, the peculiar case of the SVM classifier is significant because although its accuracy is similar to the rest of the options, its MAE value is significantly higher than the rest.

One of the possible reasons for these results may be that the dataset is highly unbalanced and the number of instances is significantly low. Certainly, both of these features are aspects to be improved in future versions.

Some of the studies previously mentioned in 1.1 that perform similar tasks to those proposed in this article [8, 9] offer results that cannot be compared with those obtained in the present study, since the systems use different datasets and the metrics used to measure their effectiveness are not the same as those used in this article to evaluate the machine learning algorithms. On the other hand, in terms of the functionality of this study with those mentioned above, it should be noted that all three provide a similar tool applied in different case studies and geographical areas. However, the system proposed in this article makes use of more accurate and reliable techniques, such as machine learning algorithms, for the prediction of the state and subsequent treatment of roads in adverse winter circumstances.

## 6 Conclusion

This paper presents a detailed multi-agent architecture designed for the accurate prediction of winter chemical treatments required for road maintenance. The proposed system demonstrates high efficiency and ensures robust communication and efficient parallelization between the different sub-objectives that are part of the system's functionality.

In particular, the paper presents a multi-agent architecture specifically designed for IoT systems, which also allows for the evolution and scalability of the system over time through a security layer that regulates the addition of new IoT devices to the platform. Another novel aspect of the architecture is the use of parallel behaviors to train various ML models, which allows the best model to be selected automatically and then used to perform predictions. The efficiency demonstrated by the different trained ML models offers a reliable system for the detection of the best chemical treatment to be applied on roads. Given these characteristics, we consider the system to be extremely beneficial and have corroborated its effectiveness using an exhaustive real case study.

However, despite the successes demonstrated, the analysis in the subsequent sections of this paper reveals that there are still opportunities for improvement. As a way forward, we propose to undertake a larger study involving a variety of classifiers and classifier ensembles to improve the predictive accuracy of the system. In addition, there is a clear need to address the current imbalance within the dataset and increase the number of data instances available, which would provide a more robust basis for analysis and decision-making.

In addition, we intend to explore strategies to reduce the dimensionality of the dataset. This reduction is expected to mitigate the impact of irrelevant or redundant data attributes that may be introducing noise into the system, thus potentially improving overall predictive

performance. With these improvements, we aim to refine the effectiveness of the system and extend its applicability to a wider range of road maintenance scenarios.

**Acknowledgements** This work was supported by the Spanish Agencia Estatal de Investigación. Project Monitoring and tracking systems for the improvement of intelligent mobility and behavior analysis (SiMoMIAC). PID2019-108883RB-C22 / AEI / 10.13039/501100011033. This work was supported by the Spanish Agencia Estatal de Investigación. Project Monitoring and tracking systems for the improvement of intelligent mobility and behavior analysis (SiMoMIAC). PID2019-108883RB-C21 / AEI / 10.13039/501100011033. The research of Diego M. Jiménez-Bravo has been co-financed by the European NextGenerationEU Fund, Spanish “Plan de Recuperación, Transformación y Resiliencia” Fund, Spanish Ministry of Universities, and University of Salamanca (“Ayudas para la recualificación del sistema universitario español para 2021-2022”).

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. European Parliament: Report on the EU Road Safety Policy Framework 2021–2030—recommendations on next steps towards ‘vision zero’ (2021/2014(INI)) (2021). [https://www.europarl.europa.eu/doceo/document/A-9-2021-0211\\_EN.html](https://www.europarl.europa.eu/doceo/document/A-9-2021-0211_EN.html)
2. Malin F, Norros I, Innamaa S (2019) Accident risk of road and weather conditions on different road types. *Accid Anal Prev* 122:181–188. <https://doi.org/10.1016/j.aap.2018.10.014>
3. El-Wakeel AS, Li J, Noureldin A, Hassanein HS, Zorba N (2018) Towards a practical crowdsensing system for road surface conditions monitoring. *IEEE Internet Things J* 5(6):4672–4685. <https://doi.org/10.1109/JIOT.2018.2807408>
4. Sattar S, Li S, Chapman M (2018) Road surface monitoring using smartphone sensors: a review. *Sensors (Switzerland)* 18(11):3845. <https://doi.org/10.3390/s18113845>
5. Chapman L, Bell SJ (2018) High-resolution monitoring of weather impacts on infrastructure networks using the internet of things. *Bull Am Meteorol Soc* 99(6):1147–1154. <https://doi.org/10.1175/BAMS-D-17-0214.1>
6. Wu M, Kwon TJ, Fu L Optimal RWIS sensor density and location—phase III: continuous mapping of winter road surface conditions via big data and deep learning. <https://trid.trb.org/view/1910740>
7. Pan G, Fu L, Yu R, Muresan M (2019) Evaluation of alternative pre-trained convolutional neural networks for winter road surface condition monitoring. In: ICTIS 2019—5th international conference on transportation information and safety, pp 614–620. <https://doi.org/10.1109/ICTIS.2019.8883540>
8. Hatamzad M, Pinerez GP, Casselgren J (2022) Addressing uncertainty by designing an intelligent fuzzy system to help decision support systems for winter road maintenance. *Safety* 8(1):14. <https://doi.org/10.3390/safety8010014>
9. Matthews L, Andrey J, Fletcher C, Oozeer Y (2021) The development of climate services to inform decisions about winter maintenance at different timescales. *Clim Serv* 22:100232. <https://doi.org/10.1016/j.cliser.2021.100232>
10. Mukherjee R, Iqbal H, Marzban S, Badar A, Brouns T, Gowda S, Arani E, Zonooz B (2021) AI driven road maintenance inspection. In: 27th ITS world congress, pp 11–15. <https://doi.org/10.48550/arxiv.2106.02567>
11. Wu S, Fang J, Zheng X, Li X (2019) Sample and structure-guided network for road crack detection. *IEEE Access* 7:130032–130043. <https://doi.org/10.1109/ACCESS.2019.2940767>
12. Dharmeeshkar J, Dhakshana S, Aniruthan S, Karthika R, Parameswaran L (2020) Deep Learning based Detection of potholes in Indian roads using YOLO. In: Proceedings of the 5th international conference on inventive computation technologies, ICICT 2020, pp 381–385. <https://doi.org/10.1109/ICICT48043.2020.9112424>

13. Alfarrarjeh A, Trivedi D, Kim SH, Shahabi C (2019) A deep learning approach for road damage detection from smartphone images. In: Proceedings—2018 IEEE international conference on big data, big data 2018, pp 5201–5204. <https://doi.org/10.1109/BigData.2018.8621899>
14. Tom RJ, Sankaranarayanan S, Rodrigues JJPC (2020) Agent negotiation in an IoT-Fog based power distribution system for demand reduction. *Sustain Energy Technol Assess* 38:100653. <https://doi.org/10.1016/j.seta.2020.100653>
15. Mostafa SA, Gunasekaran SS, Mustapha A, Mohammed MA, Abdullh WM (2020) Modelling an adjustable autonomous multi-agent internet of things system for elderly smart home. *Adv Intell Syst Comput* 953:301–311. [https://doi.org/10.1007/978-3-030-20473-0\\_29](https://doi.org/10.1007/978-3-030-20473-0_29)
16. Cagnin RL, Guilhaume IR, Queiroz J, Paulo B, Neto MFO (2018) a multi-agent system approach for management of industrial IoT devices in manufacturing processes. In: Proceedings—IEEE 16th international conference on industrial informatics, INDIN 2018, pp 31–36. <https://doi.org/10.1109/INDIN.2018.8471926>
17. Blas HSS, Mendes AS, Encinas FG, Silva LA, González GV (2021) A multi-agent system for data fusion techniques applied to the internet of things enabling physical rehabilitation monitoring. *Appl Sci (Switzerland)* 11(1):1–19. <https://doi.org/10.3390/app11010331>
18. Loureiro TCJ, Neto ABL, Rocha FAA, Aguiar FAR, Fernandez MP (2020) Multi-agent system and classification algorithms applied for health in order to support the referral of post-operative patients. *Adv Intell Syst Comput* 1006:11–18. [https://doi.org/10.1007/978-3-030-24097-4\\_2](https://doi.org/10.1007/978-3-030-24097-4_2)
19. Soon KL, Lim JMY, Parthiban R (2019) Coordinated traffic light control in cooperative green vehicle routing for pheromone-based multi-agent systems. *Appl Soft Comput J* 81:105486. <https://doi.org/10.1016/j.asoc.2019.105486>
20. Małecki K (2018) A computer simulation of traffic flow with on-street parking and drivers' behaviour based on cellular automata and a multi-agent system. *J Comput Sci* 28:32–42. <https://doi.org/10.1016/j.jocs.2018.07.005>
21. Palanisamy P (2020) Multi-agent connected autonomous driving using deep reinforcement learning. In: Proceedings of the international joint conference on neural networks. <https://doi.org/10.1109/IJCNN48605.2020.9207663>
22. Sharanappa PH, Kakkasageri MS (2022) Intelligent agents assisted information aggregation in internet of things. In: Proceedings of IEEE 2022 4th international conference on advances in electronics, computers and communications, ICAECC 2022, pp 1–7. <https://doi.org/10.1109/ICAECC54045.2022.9716588>
23. Kotenko I, Saenko I, Ageev S (2019) Applying intelligent agents for anomaly detection of network traffic in internet of things networks. In: Proceedings—2018 IEEE international conference on internet of things and intelligence system, IOTAIS 2018, pp 123–129. <https://doi.org/10.1109/IOTAIS.2018.8600867>
24. De La Iglesia DH, González GV, Mendes AS, Jiménez-Bravo DM, Barriuso AL (2019) Architecture to embed software agents in resource constrained internet of things devices. *Sensors (Switzerland)* 19(1):100. <https://doi.org/10.3390/s19010100>
25. Villarrubia G, De Paz JF, Bajo J, Corchado JM (2014) Ambient agents: embedded agents for remote control and monitoring using the PANGEA platform. *Sensors (Switzerland)* 14(8):13955–13979. <https://doi.org/10.3390/s140813955>
26. Agüero J, Rebollo M, Carrascosa C, Julián V (2012) Towards on embedded agent model for Android mobiles. <https://doi.org/10.4108/icst.mobiquitous2008.4195>
27. Doctor F, Hagraas H, Callaghan V (2005) A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments. *Inf Sci* 171(4):309–334. <https://doi.org/10.1016/j.ins.2004.09.008>
28. Telecom Italia S.p.A.: Jade Site Java Agent DEvelopment Framework (2017). <http://jade.tilab.com/>
29. Braubach L, Pokahr A, Lamersdorf W (2005) Jadex: a BDI-agent system combining middleware and reasoning. *Software agent-based applications, platforms and development kits*, pp 143–168. [https://doi.org/10.1007/3-7643-7348-2\\_7](https://doi.org/10.1007/3-7643-7348-2_7)
30. sphinx-quickstart: Python Agent DEvelopment framework—Pade 1.0 documentation (2015). <https://pade.readthedocs.io/en/latest/>
31. Palanca J, Terrasa A, Julian V, Carrascosa C (2020) Spade 3: supporting the new generation of multi-agent systems. *IEEE Access* 8:182537–182549. <https://doi.org/10.1109/ACCESS.2020.3027357>
32. osBrain—0.6.5—osBrain 0.6.5 documentation. <https://osbrain.readthedocs.io/en/stable/>
33. C4.5 (1993). <https://doi.org/10.1016/C2009-0-27846-9>
34. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32. <https://doi.org/10.1023/A:1010933404324>
35. Cohen WW (1995) Fast effective rule induction. *Mach Learn Proc* 1995:115–123. <https://doi.org/10.1016/b978-1-55860-377-6.50023-2>
36. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. *Mach Learn* 6(1):37–66. <https://doi.org/10.1007/bf00153759>

37. Boser BE, Guyon IM, Vapnik VN (1992) Training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual ACM workshop on computational learning theory, pp 144–152. <https://doi.org/10.1145/130385.130401>
38. John GH, Langley P (2013) Estimating continuous distributions in Bayesian classifiers. <https://doi.org/10.48550/arxiv.1302.4964>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Diego M. Jiménez-Bravo** studied Computer Engineering for Management and Information Systems at the University of the Basque Country (UPV/EHU). Subsequently, he obtained a Master's degree in Intelligent Systems from the University of Salamanca (USAL). He concluded his formative stage by obtaining a PhD in Computer Engineering from the USAL. He is part of the ESALab research group and the Department of Computer Science and Automation of the same entity, where he carries out teaching and research work through a contract of Associate Professor in Computer Science and Artificial Intelligence. He has previously been awarded various postdoctoral research contracts funded by national and European institutions. His main research interests are in the field of artificial intelligence, IoT (Internet of Things), smart-homes, energy optimization, and social networks, among others. In addition, he collaborates in several research projects within his research group.



**Javier Bajo** full professor at the Department of Artificial Intelligence, Computer Science School at Universidad Politécnica de Madrid (UPM) and Director of the UPM AI.nnovation Space Research Center in Artificial Intelligence. His main lines of research are Social Computing and Artificial and Hybrid Societies; Intelligent Agents and Multiagent Systems, Ambient Intelligence, Machine Learning. He has supervised 15 Ph.D. thesis, participated in more than 50 research projects (in most of them as principal investigator), and published more than 300 articles in recognized journals (81 JCR papers).



**Jacinto González-Pachón** is Associate Professor in the Department of Artificial Intelligence at Universidad Politécnica de Madrid (UPM). He holds a B.S in Mathematics from Universidad de Extremadura and a Ph.D. in Computer Science from UPM. His teaching and research specialities are in the field of preference modelling and rational choice, multi-criteria decision-making problems, group decision-making problems and uncertainty in decision analysis. He is the author of papers that have appeared in international refereed journals and monographs such as *Computers and Operation Research*, *European Journal of Operational Research*, *Journal of the Operational Research Society*, *Lecture Note in Economics and Mathematical Systems*, *OMEGA*, *Theory and Decision*, and others.



**Juan F. De Paz** holds degrees in technical engineering in computer systems, engineering in computer science, and statistics, as well as a Ph.D. degree in computer science from the University of Salamanca, Spain, earned in 2003, 2005, 2007, and 2010, respectively. He currently serves as a Full Professor at the University of Salamanca and is an active Researcher with the Expert Systems and Applications Laboratory (ESALab). Juan has co-authored numerous published papers in various journals, workshops, and symposiums.