

API serveur de prédiction

Description

1. Lecteurs ciblés

Cette documentation est à destination des utilisateurs du serveur de prédiction, permettant la génération de modèles d'intelligence artificielle et la prédiction de données.

2. Présentation de l'API

Le serveur de prédiction a été créé dans le cadre du projet de recherche PREDIT READYNNOV. Son but est de créer et d'améliorer des modèles d'intelligence artificielle parcimonieux en se basant sur le logiciel NeurEco développé par notre partenaire Adagos. Les modèles créés sont des modèles de prédiction de la production photovoltaïque ou de prédiction de la consommation notamment. Grâce à ces modèles, le serveur peut prévoir des données par exemple la consommation ou la production photovoltaïque d'une installation sur une semaine.

3. Fonctionnalités

3.1 Headers

Information
Fonctionnalité non encore implémentée.

3.2 Méthode GET

Grâce à l'API du serveur, il est possible d'obtenir des informations sur:

- l'état d'apprentissage du modèle
- les prédictions calculées par le modèle
- la date de dernière requête météo prévisionnelle pour un modèle

3.2.1 État d'apprentissage

Pour obtenir l'état d'apprentissage d'un modèle, la requête HTTP sera:

/learning?

Cette requête sera suivie d'un seul paramètre:

id=VALUE Index (VALUE) du modèle

Tableau sur le paramètre d'une requête GET learning

La réponse à la requête sera un dictionnaire constitué des clés/valeurs suivantes:

Nom du champs	Description
step	Étape à laquelle se situe la génération du modèle
nbSteps	Nombre d'étapes lors de la génération du modèle
startTime	Date au format YYYY-MM-DDThh:mm:ss.f à laquelle la génération du modèle a commencé

Un exemple de requête HTTP est :

`https://predict.sirea.fr/learning?id=1_1671022828`

La réponse à cette requête sera:

```
{"step": 0, "nbSteps": 2, "startTime": 0}
```

Un autre exemple de réponse sera:

```
{"step": 1, "nbSteps": 2, "startTime": "2022-12-29T16:04:37.643461"}
```

Si le champ 'step' est à 0 cela signifie que la génération du modèle n'a pas commencé.

Si le champ 'step' est à 1 cela signifie que le modèle est en cours de génération.

Si le champ 'step' est à 2 cela représente que la génération du modèle est terminée sans aucune erreur.

Si le champ 'step' est à -1 cela veut dire que la génération du modèle ne s'est pas terminée pour cause d'erreur.

Information

La génération d'un modèle peut être assez longue et prendre quelques heures lo

3.2.2 Prédictions

Pour obtenir les prédictions calculées par un modèle sur une ou des dates précises, la requête HTTP sera:

`/prediction?`

Cette requête sera suivie de plusieurs paramètres listés ci-dessous:

<code>id=VALUE</code>	Index (VALUE) du modèle
<code>time=VALUE</code> ou <code>time[]</code> <code>=VALUES</code>	Valeur (VALUE) du timestamp pour lequel on souhaite une prédiction. On peut aussi avoir une liste de timestamps (VALUES)
<code>weatherPrecision=VALUE</code>	Précision des prédictions. VALUE peut être égal à 'hour' (précision à l'heure) ou 'day' (précision au jour)

Tableau sur le paramètre d'une requête GET prediction

Un exemple de requête HTTP est :

`https://predict.sirea.fr/prediction?id=1_1667223889&weatherPrecision=hour&time`

La réponse à cette requête sera:

`[[133.64790489627035], [null], [null]]`

La réponse à une requête de prédiction sera une liste de listes de valeurs. Chaque sous-liste *i* contient la/les valeurs de prédiction du timestamp envoyé au rang *i*.

Information

Il est à noter que dans la réponse précédente à la requête, on observe des val



Capture d'écran de la production photovoltaïque réelle et prédite pour une installation

3.2.3 Date de dernière requête météo prévisionnelle

Pour obtenir la date de dernière requête météo prévisionnelle faite par le serveur, la requête HTTP sera:

`/weather-data-time?`

Cette requête sera suivie d'un paramètre:

`id=VALUE` Index (VALUE) du modèle

Tableau sur le paramètre d'une requête GET weather-data-time

Un exemple de requête HTTP est :

`https://predict.sirea.fr/weather-data-time?id=1_1667223889`

La réponse à cette requête sera:

`"2022-12-28 16:22:26"`

Information

Il est à noter que la réponse est une chaîne de caractère contenant une date a

3.2.4 Bloc fonction de prédiction

Pour obtenir le bloc fonction permettant de réaliser des requêtes au serveur, la requête HTTP sera:

`/ladder-file`

Cette requête n'est suivie d'aucun paramètre.

Ainsi, la requête entière sera:

`https://predict.sirea.fr/ladder-file`

La réponse de cette requête sera la téléchargement automatique du bloc fonction dans le dossier « Téléchargements » de votre ordinateur.

3.3 Méthode POST

Cette méthode est utilisée pour créer un nouveau modèle encore vierge c'est à dire qui n'a jamais appris et qui ne peut donc encore rien prévoir (en réalité se sont les dossiers et fichiers qui gèrent le modèle qui vont être créés).

Pour créer un nouveau modèle, la requête HTTP sera:

`/learning`

Cette requête sera potentiellement suivie d'un dictionnaire. Ainsi, on peut renseigner avec des valeurs les clés suivantes:

Nom du champs	Description
useDate	True/False, booléen indiquant si le modèle apprendra avec des données temporelles en entrées
weatherData	Dictionnaire composé des paramètres météo (voir tableau suivant pour plus de renseignements)
values	Liste de dictionnaire des valeurs d'entrées et de sorties pour la génération d'un modèle

Tableau sur les champs disponibles pour la création d'un modèle

La clé 'values' est une liste composée de dictionnaires dont la trame est la suivante:

```
{"time": ..., "input": [...], "output": [...]}
```

Un exemple est:

```
"values": [{"time": 1577836800, "input": [], "output": ["0.45893"]}, {"time":
```

Si cette clé/valeur est présente alors la génération du modèle va se lancer.

Nom du champs	Description
latitude	Valeur de la latitude où se situe l'installation
longitude	Valeur de la longitude où se situe l'installation
parameters	Liste des paramètres météo à prendre en compte lors de la génération du modèle. Les paramètres disponibles sont: solarIrradiance, airTemperature, cloudCover, airHumidity, windSpeed et windDirection
precision	hour/day, chaîne indiquant le degrés de précision de l'apprentissage du modèle

Tableau sur les clés disponibles pour le dictionnaire weatherData

Paramètres	Description
solarIrradiance	Irradiance normale horizontale
airTemperature	Température de l'air
cloudCover	Couverture nuageuse
airHumidity	Humidité ambiante de l'air
windSpeed	Vitesse du vent
windDirection	Direction du vent

Tableau sur la description des paramètres météo

Information

Il est à noter que le modèle sera identifié par un index propre à lui-même. Ce

Un exemple de requête HTTP est:

<https://predict.sirea.fr/learning>

avec comme paramètres:

```
{"useDate": true, "weatherData": {"latitude": "43.5", "longitude": "2.4", "par
```

3.4 Méthode PATCH

Cette méthode est utilisée pour modifier un modèle.

3.4.1 Génération d'un modèle

Pour générer un modèle, la requête HTTP sera:

```
/learning/{id}
```

Avec 'id' correspondant à l'index du modèle. Cette requête sera suivie d'un dictionnaire dont les spécificités sont décrites dans le tableau suivant:

Nom du champs	Description
useDate	True/False, booléen indiquant si le modèle apprendra avec des données temporelles en entrées
weatherData	Dictionnaire composé des paramètres météo (voir tableau suivant pour plus de renseignements)
values	Liste de dictionnaire des valeurs d'entrées et de sorties pour la génération d'un modèle

Tableau sur les champs disponibles pour la création d'un modèle

La clé 'values' est une liste composée de dictionnaires dont la trame est la suivante:

```
{"time": ..., "input": [...], "output": [...]}
```

Un exemple est:

```
"values": [{"time": 1577836800, "input": [], "output": ["0.45893"]}, {"time":
```

Nom du champs	Description
latitude	Valeur de la latitude où se situe l'installation
longitude	Valeur de la longitude où se situe l'installation
parameters	Liste des paramètres météo à prendre en compte lors de la génération du modèle. Les paramètres disponibles sont: solarIrradiance, airTemperature, cloudCover, airHumidity, windSpeed et windDirection
precision	hour/day, chaîne indiquant le degrés de précision de l'apprentissage du modèle

Tableau sur les clés disponibles pour le dictionnaire weatherData

Paramètres	Description
solarIrradiance	Irradiance normale horizontale
airTemperature	Température de l'air
cloudCover	Couverture nuageuse
airHumidity	Humidité ambiante de l'air
windSpeed	Vitesse du vent
windDirection	Direction du vent

Tableau sur la description des paramètres météo

Un exemple de requête HTTP est:

`https://predict.sirea.fr/learning/1_1667223889`

avec comme paramètres:

`{"useDate": true, "weatherData": {"latitude": "43.5", "longitude": "2.4", "par`

3.4.2 Amélioration d'un modèle

Une fois qu'un modèle a été généré, il est possible de l'améliorer en lui donnant de nouvelles données sur lesquelles lancer un nouvel apprentissage.

La requête HTTP sera:

`/learning/{id}`

Ce qui va différencier cette requête de la précédente est que le dictionnaire suivant cette requête est composé de clé/valeur différentes:

Nom du champs	Description
extraValues	Liste de dictionnaire des valeurs d'entrées et de sorties pour l'amélioration d'un modèle

Tableau sur les champs disponibles pour la création d'un modèle

Un exemple de requête HTTP est:

`https://predict.sirea.fr/learning/1_1667223889`

avec comme paramètres:

`{"extraValues": [{"time": 1577836800, "input": [], "output": ["0.45893"]}], {"t`

3.5 Méthode DELETE

Cette méthode est utilisée pour supprimer un modèle.

Ainsi pour supprimer un modèle la requête sera:

```
/learning/{id}
```

Avec 'id' correspondant à l'index du modèle. Il n'y a besoin d'aucun paramètre.

3.6 Erreurs

3.6.1 Mauvaise requête

Si la requête saisie n'existe pas alors le serveur va renvoyer une erreur 400 Bad request ainsi que la réponse:

```
{"message": "Wrong request"}
```

3.6.2 Paramètre manquant

Si un paramètre de la requête est manquant (par exemple l'index du modèle 'id') alors le serveur va renvoyer une erreur 400 Bad request ainsi que la réponse:

```
{"message": "Missing parameters"}
```