



مدين المهن و الكفاءات  
+٤٣٦٨٤١٤١ | +٠٥٢٩٦٤١٨٥٠  
Cités des Métiers et des Compétences

# Rapport de Projet Final

**Centre de Commandement et d'Analyse Stratégique :**

**Un Système de Détection de Menaces Globales Basé sur le Machine Learning**

<b>Stagiaire :</b>	Oussama Aslouj
<b>Année :</b>	2024/2025
<b>Filière :</b>	Intelligence Artificielle
<b>Groupe :</b>	104
<b>Établissement :</b>	OFPPT - Cité des Métiers et des Compétences, Casablanca-Settat
<b>Formateur :</b>	M. EL MOUDEN ABDELAZIZ
<b>Date :</b>	19 Juin 2025

## B. REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin, à la réussite de ce projet.

Mes remerciements s'adressent tout particulièrement à mon formateur, **M. EL MOUDEN ABDELAZIZ**, pour son encadrement précieux, sa patience, sa disponibilité et ses conseils avisés qui ont guidé ma réflexion tout au long de ce travail.

Je remercie également l'**Office de la Formation Professionnelle et de la Promotion du Travail (OFPPT)**, ainsi que la direction et l'ensemble du corps professoral de la **Cité des Métiers et des Compétences (CMC) de la région Casablanca-Settat**, pour la qualité de la formation dispensée qui a été le socle de ce projet.

Enfin, je n'oublie pas mes collègues pour leur soutien et l'esprit d'entraide qui a régné durant notre formation.

## C. Résumé du projet

Ce projet consiste en la conception et le développement d'une application de bureau simulant un "Centre de Commandement" pour la surveillance de menaces globales. L'objectif principal était de mettre en œuvre un cycle complet de développement en apprentissage automatique pour analyser et classifier différentes menaces : sismiques (explosions nucléaires, tsunamis), solaires, orbitales (astéroïdes) et humanitaires. En utilisant des techniques variées telles que la détection d'anomalies (IsolationForest) et la classification supervisée (RandomForestClassifier), le projet intègre de multiples API publiques (USGS, NASA, GDACS) pour alimenter un dashboard interactif en temps réel, démontrant ainsi une solution complète allant de la collecte de données au déploiement d'un modèle dans une application fonctionnelle.

### Introduction générale

#### Contexte général

À l'ère de l'information, la capacité à agréger, analyser et réagir à des événements globaux est devenue un enjeu stratégique majeur. Les menaces, qu'elles soient d'origine naturelle ou humaine, nécessitent des outils de surveillance performants pour anticiper les risques et coordonner les réponses. Ce projet s'inscrit dans ce contexte en proposant une solution centralisée, un "Centre de Commandement", qui utilise la puissance de l'apprentissage automatique pour fournir une analyse multi-domaines des menaces.

#### Objectifs du projet

L'objectif général est de réaliser un mini-projet complet en apprentissage automatique, simulant une problématique réelle. Les objectifs spécifiques étaient :

- Développer et entraîner plusieurs modèles de Machine Learning (supervisé et non supervisé) pour différentes tâches de détection.
- Intégrer ces modèles dans une application fonctionnelle avec une interface graphique moderne et interactive.
- Manipuler et traiter des données en temps réel provenant de multiples API publiques.
- Implémenter des logiques d'analyse expertes (estimation de puissance, score de risque, etc.) pour enrichir les prédictions des modèles.

#### Méthodologie adoptée

Le projet a suivi le cycle de vie d'un projet de machine learning :

1. **Définition de la problématique** : Créer un système de surveillance multi-menaces.
2. **Collecte des données** : Utilisation de jeux de données historiques (earthquake.csv, tsunami\_dataset.csv) pour l'entraînement et d'API pour les données en direct.
3. **Prétraitement des données** : Nettoyage, sélection de caractéristiques pertinentes et ingénierie des variables cibles.
4. **Modélisation** : Entraînement de deux modèles distincts et sauvegarde avec joblib.
5. **Déploiement et Évaluation** : Intégration des modèles dans une application CustomTkinter et évaluation qualitative des performances en temps réel.

## Partie 1: Fondements théoriques

- **Apprentissage Non Supervisé - Détection d'Anomalies** : Le cœur du module sismique repose sur l'algorithme IsolationForest. Ce dernier isole les observations en sélectionnant aléatoirement une caractéristique puis en choisissant une valeur de division aléatoire. Les anomalies, étant "peu nombreuses et différentes", sont plus rapides à isoler. Nous l'avons entraîné sur des données de séismes "normaux" pour qu'il identifie toute signature anormale (explosion) comme une anomalie.
- **Apprentissage Supervisé - Classification** : Pour la prédiction de tsunamis, une approche supervisée avec l'algorithme RandomForestClassifier a été utilisée. C'est un modèle ensembliste qui construit une multitude d'arbres de décision lors de l'entraînement et produit la classe qui est le mode des classes des arbres individuels. Il a été entraîné sur une base de données historique pour apprendre la relation entre la magnitude/profondeur d'un séisme et la génération d'un tsunami.
- **Seuils de Décision et Probabilités** : Pour affiner la sensibilité des modèles, nous n'utilisons pas seulement la prédiction brute (0 ou 1), mais aussi le score de confiance (decision\_function pour IsolationForest) et la probabilité (predict\_proba pour RandomForest), nous permettant d'instaurer des niveaux d'alerte nuancés.

## Partie 2: Outils et technologies utilisés

- **Python** : Langage principal pour sa polyvalence et son écosystème.
- **Scikit-learn** : Pour l'implémentation des modèles IsolationForest et RandomForestClassifier et des outils de prétraitement (StandardScaler).

- **Pandas** : Pour la manipulation et le nettoyage des jeux de données.
- **CustomTkinter** : Pour la création de l'interface graphique de bureau moderne et réactive.
- **TkinterMapView** : Pour l'intégration de la carte interactive.
- **Requests & Feedparser** : Pour interroger les API REST et les flux RSS en temps réel.
- **Shapely** : Pour la gestion des zones géopolitiques (polygones).

### Partie 3: Préparation des données

Deux jeux de données principaux ont été utilisés pour l'entraînement :

1. **earthquake.csv** : Contenant des données sur des séismes et des explosions, il a été filtré pour ne garder que les séismes "normaux" afin d'entraîner le modèle de détection d'anomalies.
2. **tsunami\_dataset.csv** : Une base de données historique d'événements sismiques. Une variable cible binaire (TSUNAMI\_EVENT) a été créée à partir de la colonne TS\_INTENSITY pour l'entraînement du modèle supervisé.

Les deux jeux de données ont nécessité une sélection des caractéristiques (Magnitude, Depth, etc.) et un nettoyage pour gérer les valeurs manquantes.

### Partie 4: Modélisation et évaluation

- **Modèle 1 (Anomalie Sismique)** : Un IsolationForest a été entraîné. Son évaluation est qualitative : via le mode "Time Machine", nous avons confirmé sa capacité à identifier l'essai nucléaire nord-coréen de 2017 comme une anomalie critique, validant son efficacité.
- **Modèle 2 (Tsunami)** : Un RandomForestClassifier a été entraîné. L'évaluation via classification\_report a montré une bonne performance. Pour éviter une sur-sensibilité, une double condition a été ajoutée dans l'application : probabilité > 70% ET magnitude > 7.5.
- **Optimisation et Recommandations** : L'optimisation principale n'a pas été sur les hyperparamètres, mais sur la logique d'application des modèles. L'ajout de règles expertes (zones de risque, seuils de magnitude, estimation de puissance) constitue la principale recommandation pour améliorer la pertinence des alertes.

### Partie 5: Intégration et déploiement du modèle

Le projet a abouti à une application de bureau complète et fonctionnelle.

- **Interface** : Une fenêtre principale organisée en quatre onglets, construite avec CustomTkinter.

- **Visualisation Centrale** : Une carte interactive (TkinterMapView) affiche les événements sismiques en temps réel avec des marqueurs animés, colorés et cliquables.
- **Dashboard** : Les onglets Solaire et Astéroïdes présentent des cartes de synthèse qui résument les informations critiques, en plus des journaux détaillés.
- **Interactivité** : Le mode "Time Machine", les fenêtres de détails et les boutons d'exportation CSV rendent l'application pleinement interactive.

### Difficultés rencontrées & solutions

- **Fiabilité des Données** : Trouver des jeux de données propres et bien étiquetés était un défi. La solution a été de nettoyer et de préparer nous-mêmes les données brutes (par ex., créer la cible pour le modèle tsunami).
- **Bugs des Bibliothèques** : Des incompatibilités et des fonctions non documentées dans les bibliothèques graphiques (tkintermapview) ont causé des erreurs. La solution a été de déboguer pas à pas et de trouver des solutions de contournement robustes.
- **Gestion des Alertes** : Les modèles étaient initialement trop sensibles. La solution a été d'implémenter des seuils de confiance et des règles expertes pour réduire les "fausses" alertes et rendre le système plus fiable.

### Conclusion générale

Ce projet a été une réussite complète, dépassant les objectifs initiaux. Il a permis de mettre en pratique l'ensemble du cycle de vie d'un projet de Machine Learning, de l'idéation à la création d'une application complexe et fonctionnelle.

### Compétences acquises :

- Maîtrise de modèles supervisés et non supervisés.
- Ingénierie de caractéristiques et traitement de données complexes.
- Intégration de multiples API en temps réel.
- Développement d'interfaces graphiques avancées avec Python.
- Gestion de projet et résolution de problèmes complexes.

### Perspectives d'amélioration :

- Entraîner les modèles sur des jeux de données encore plus vastes.
- Ajouter un module de prédiction météorologique pour les cyclones.
- Créer une version web de l'application pour une accessibilité universelle.