Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Hes·so

Etudiant :
Adam Grossenbacher

Résumé travail de Bachelor 2024
Informatique et systèmes de communication

Professeurs : Jean Hennebert , Beat Wolf

Mandant : LYSR sàrl

Contact Interne : iCoSys

Filière : Informatique et systèmes de communication

Orientation : Ingénierie des données

Experts : Gérôme Bovet , Geoffrey Papaux

Objectifs de dév. durable :

SYSTÈME DE BACKTESTING HÉBERGÉ SUR KUBERNETES POUR LA DÉTECTION D'ANOMALIES

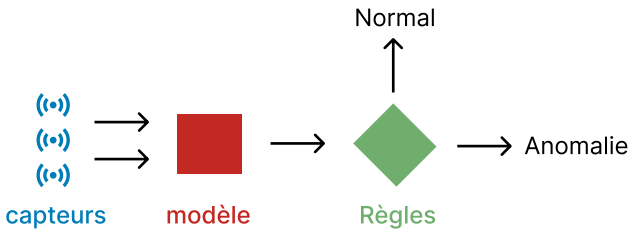
Ce travail de bachelor, réalisé à la Haute école d'ingénierie et d'architecture de Fribourg (HEIA-FR) en collaboration avec l'institut iCoSys et la startup LYSR, a pour objectif de développer un système de backtesting pour évaluer des modèles de détection d'anomalies basés sur l'IA à partir de données historiques. Le projet comprend le développement d'un prototype de détection d'anomalies, la mise en place d'un système de backtesting qui utilise des technologies telles que Ray, Kubernetes et MLflow, ainsi que son intégration au backend et au frontend de la plateforme LYSR.

Anomalies

Une anomalie est un écart par rapport à la norme ou à une valeur théorique. Il existe plusieurs types d'anomalies, telles que les anomalies ponctuelles, collectives et contextuelles. La détection d'anomalies, essentielle dans divers secteurs tels que la finance, la fabrication et la cybersécurité, vise à identifier ces écarts pour améliorer les performances et prévenir d'éventuels dysfonctionnements. Les méthodes de détection peuvent être manuelles ou basées sur l'intelligence artificielle (IA), comme dans ce travail de bachelor. L'évaluation des modèles d'IA pour la détection d'anomalies utilise des métriques telles que la précision, le rappel, le F1 score et l'AUC. Les défis incluent la définition des anomalies, le déséquilibre des classes et la qualité des données.

LYSR

La startup LYSR, spécialisée dans la détection d'anomalies et la maintenance prédictive des données de séries temporelles pour l'industrie 4.0, propose une plateforme fiable et facile à utiliser pour stocker, visualiser et analyser ces données afin d'y détecter des anomalies. Le graphique ci-dessous schématise le fonctionnement de LYSR pour détecter des anomalies.

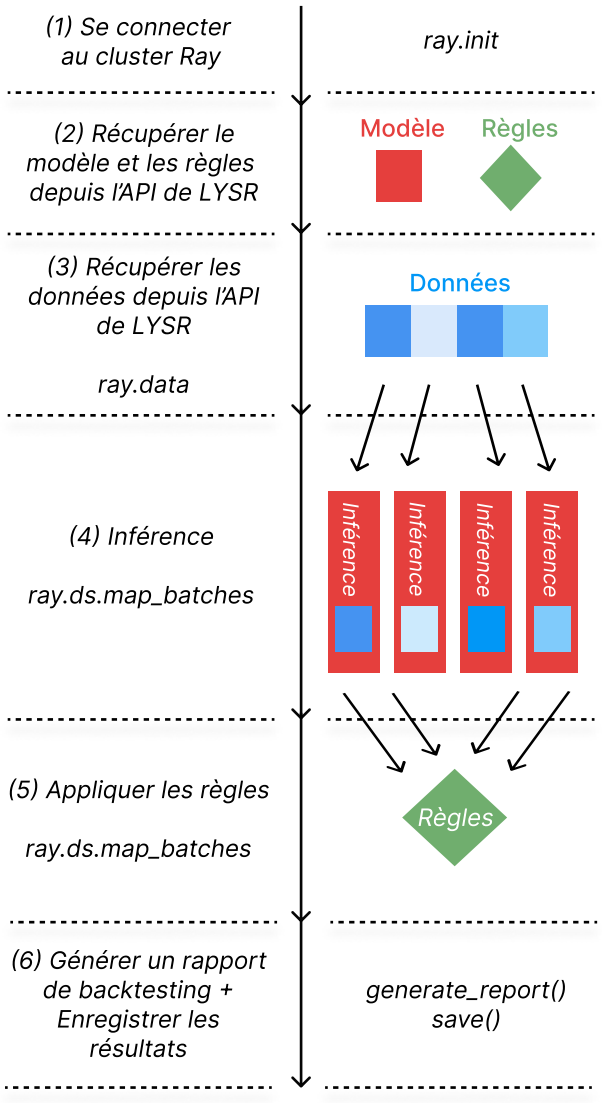


Les données sont générées par des capteurs placés sur des machines industrielles, qui envoient des données en temps réel. Ces données sont ensuite analysées par un modèle d'IA qui retourne différentes valeurs. Enfin, des règles sont appliquées aux valeurs retournées par le modèle pour déterminer s'il s'agit d'une anomalie ou non.

Backtesting

Le backtesting consiste à évaluer les performances d'un modèle d'IA avec des données historiques, un aspect crucial avant l'utilisation en production. Actuellement, LYSR effectue le backtesting en local, ce qui complique le suivi et la reproductibilité. Elle souhaite intégrer un système de backtesting à sa plateforme, d'où ce mandat de bachelor.

Système de backtesting distribué avec Ray
Le système de backtesting développé repose sur Ray, un framework de calcul distribué qui lui-même repose sur Kubernetes. Ray permet d'exécuter de manière scalable et efficace diverses tâches de machine learning, telles que l'entraînement de modèles d'IA ou l'inférence, de manière distribuée. Un cluster Ray comprend plusieurs workers exécutant ces tâches, y compris celles du backtesting développé dans ce projet. Le script Python initiant le backtesting est schématisé ci-dessous.



Chaque tâche est effectuée sur le cluster Ray. Les tâches (3), (4) et (5) utilisent ray.data pour découper les données en blocs et les traiter avec map_batches, permettant une inférence et une application des règles de manière distribuée ou parallélisée, accélérant ainsi le processus de backtesting et permettant de gérer une grande charge de travail.

Système de backtesting : autres éléments
Outre le script de backtesting, plusieurs nouvelles fonctionnalités ont été développées :

- Un script Python génère un rapport de backtesting au format HTML, facilitant ainsi la compréhension et l'interprétation des résultats. Ce rapport inclut des matrices de confusion, des métriques telles que la précision, le rappel et le score F1, ainsi que des graphiques de séries temporelles.
- Une API Rest en Java permet d'initier le backtesting sur le cluster Ray, de suivre son état, d'afficher les logs, de récupérer les résultats et le rapport de backtesting.
- Un serveur de fichiers permet de récupérer le script de backtesting depuis n'importe quel worker.
- Un système de stockage d'objets (MinIO) regroupe les résultats du backtesting au format Parquet et le rapport au format HTML.
- Une interface web permet de visualiser les modèles, règles et données, de sélectionner les éléments à tester et de lancer le backtesting. Le rapport et les logs peuvent également être consultés.

Résultats
Les résultats démontrent que le système de backtesting fonctionne efficacement et s'intègre bien à la plateforme LYSR, répondant ainsi aux objectifs fixés. Des améliorations futures sont proposées pour optimiser l'efficacité et enrichir les fonctionnalités.

Durabilité
La détection d'anomalies contribue aux Objectifs de Développement Durable (ODD). Elle aide à une consommation et une production responsables (réduction du gaspillage), améliore la santé et le bien-être (diagnostic précoce), soutient des villes durables (prévention des pannes), et lutte contre le changement climatique (alerte précoce). Cependant, elle n'aborde pas tous les ODD et pose des défis environnementaux liés aux matériaux rares, aux déchets électroniques et à la consommation d'énergie. Le système de backtesting développé optimise légèrement la consommation énergétique grâce à Ray et Kubernetes, qui réduisent les pics de consommation en répartissant efficacement la charge.