Détection de Singularités en temps-réel par combinaison d'apprentissage automatique et web sémantique basés sur Spark

Badre Belabbess[*,**] Musab Bairat [**] Jeremy Lhez [*]
Olivier Curé [**]

[*]Innovation Lab, ATOS, F-95870, Bezons, France
prénom.nom@atos.net,
[**]LIGM (UMR 8049), CNRS, F-77454, MLV, France.
prénom.nom@univ-paris-est.fr

1 Introduction

L'apprentissage automatique contient un ensemble puissant d'approches qui peut aider à détecter des anomalies de manière efficace. Cependant, il représente un processus lourd avec des règles strictes et une multitude de tâches telles que l'analyse et le nettoyage des données, la réduction de dimension, l'échantillonnage, la sélection d'algorithmes appropriés, le réglage précis des hyper-paramètres, etc. Notre système a été spécifiquement conçu pour simplifier ce processus lourd et accélérer le déploiement d'une solution en peu de temps. Notre système vise à identifier les anomalies dans un grand réseau d'eau potable géré par un leader national expert dans le domaine de l'eau. En fait, la découverte de telles irrégularités dans le réseau d'eau est une préoccupation critique tant sur le plan écologique que financier. Le volume réel d'eau perdue dans le monde a généré une perte de 32 milliards de m3 / an (soit 14 milliards d'euros par an) dont 90 % reste difficilement identifiable en raison de la nature souterraine du réseau. Sur la base de recherches approfondies menées par les experts, les anomalies peuvent être identifiées en utilisant des mesures de pression et de débit envoyées par des capteurs spécifiques dispersés sur tout le réseau de canalisations.

2 Architecture

Le système a été conçu pour traiter à la fois des données massives dynamiques et statiques à l'aide d'une architecture distribuée tolérante aux pannes. L'objectif principal est de pouvoir traiter des flux massifs de données en temps réel et de lancer des modèles intensifs d'apprentissage automatique. Pour répondre aux besoins d'un système distribué robuste, scalable et à faible latence, nous avons basé notre conception sur une architecture Lambda. Ce type d'architecture Big Data résout le problème des fonctions de calcul lourdes sur des données en temps réel en décomposant le problème en trois couches : une couche batch, une couche vitesse et couche service. Un scénario général de bout en bout commence par le stockage des données

historiques horodatées sous forme de séries-temporelles à des fins de pré-analyse. La mise en cache de données massives nécessite un système de fichiers distribué robuste pour récupérer les données très rapidement. Le système utilise un cluster Hadoop lors de la première phase de traitement en batch. Cependant, dans la plupart des cas, les données brutes doivent être nettoyés pour augmenter la précision de l'identification des singularités. Deux étapes se succèdent ici, le système infère les données manquantes via des techniques d'interpolation et de maximisation de l'espérance, ensuite des techniques de réduction de dimensions permettent de réduire la taille du dataset initial. Une unité de modélisation distribuée appliquera plusieurs modèles de séries temporelles pour trouver des valeurs aberrantes (*e.g.*, saisonnalité, chronologie, profil, etc). Les valeurs aberrantes seront utilisées pour classer les attributs selon la probabilité d'occurrence d'anomalie, les modèles n'étant appliqués que sur les attributs les mieux classés réduisant ainsi considérablement le temps de traitement. Cette méthode permet une allocation de données dynamique en optimisant la taille des paquets de données transférés entre le HDFS et le moteur Spark. Cette allocation de données est gérée par une unité sémantique profitant des atouts des ontologies. Après avoir converti les données réduites en RDF, un générateur de requêtes continues en SPARQL sélectionnera le graphique de taille minimale en utilisant une ontologie conçue pour le cas d'utilisation actuel. Afin de sélectionner l'algorithme correspondant au profil des flux ingérés, nous utilisons un ensemble complexe de règles telles que l'interdépendance des variables, le profil de distribution des données ou l'estimation de la complexité du traitement. Les résultats trouvés seront envoyés vers un système de messagerie, Apache Kafka, qui mettra en file d'attente les messages de manière ordonnée pour être exposés par un outil de visualisation. Enfin, le système s'appuie sur les annotations de l'utilisateur final pour lancer une nouvelle boucle d'itération qui stockera les signatures de chaque anomalie validée.

3 Contributions

Le système proposé est évolutif, permettant la détection d'anomalies sur des flux en temps réel à l'aide d'un mélange de techniques d'apprentissage automatique et d'une approche web sémantique. En s'appuyant sur le profil des données historiques, le système utilise un ensemble de règles hiérarchiques pour sélectionner le meilleur algorithme adapté au cas d'usage. Nous estimons qu'il s'agit du premier système visant à automatiser le processus complet d'apprentissage automatique depuis le nettoyage des données au lancement des modèles appropriés. Les tâches effectuées au cours de processus sont généralement effectuées manuellement par des experts qui ont besoin d'une connaissance suffisante du domaine de d'application et de l'apprentissage automatique en général. Utilisant les capacités d'un environnement distribué pour traiter des données massives et véloces, notre système propose de trouver automatiquement l'algorithme pouvant les meilleurs résultats en terme de précision et de temps d'exécution.

Summary

Using machine learning to solve complex use cases is generally a cumbersome, costly, and error-prone process. With our system, we remove the burden of this process and demonstrate that many machine learning tasks can be automated.