ENSAE PARIS - INSTITUT POLYTECHNIQUE DE PARIS

MILLIMAN FRANCE

Projet de Statistiques Appliquées - Note de Synthèse

Apports et perspectives du Federated Learning dans le domaine assurantiel

Mai 2025

AUTEURS :
ABE Kevin
BENABDESADOK Nayel
HOUNKPEVI Crespin
REN Alexandre

SUPERVISÉ PAR :
François HU, Head of AI Lab chez Milliman
Fallou NIAKH, Encadrant de la voie Actuariat à l'ENSAE Paris
Caroline HILLAIRET, Professeure et Responsable de la voie Actuariat

Table des matières

1 Introduction et Motivation

Dans le cadre de ce projet, nous nous intéressons à l'analyse de la sinistralité automobile à l'échelle européenne, avec pour objectif de construire un modèle prédictif capable d'estimer la probabilité de survenue d'un sinistre pour un assuré. Cette démarche repose sur la collaboration entre plusieurs compagnies d'assurance implantées dans différents pays (France, Belgique, Espagne, etc.), chacune fournissant des données propres à sa clientèle.

Une stratégie évidente consisterait à centraliser toutes ces données pour entraîner un modèle global. Toutefois, cette approche se heurte à deux obstacles majeurs : d'une part, l'hypothèse d'indépendance et d'identique distribution des données (i.i.d.) est invalidée par l'hétérogénéité des clientèles entre assureurs ; d'autre part, la centralisation de données sensibles est fortement limitée par des contraintes réglementaires, notamment le Règlement Général sur la Protection des Données (RGPD).

Pour répondre à ces défis, notre projet s'appuie sur un paradigme émergent : l'apprentissage fédéré (*Federated Learning*). Cette méthode permet à plusieurs entités de participer à l'entraînement d'un modèle commun sans échanger leurs données locales, garantissant ainsi la confidentialité tout en profitant de l'information distribuée.

Déjà utilisé dans des domaines comme la santé ou la cybersécurité, le FL présente un fort potentiel pour le secteur assurantiel, où les données sont à la fois sensibles et hétérogènes (diagnostics médicaux, historique des sinistres, infractions,...).

2 Principes Généraux de l'Apprentissage Fédéré

Le principe de l'Apprentissage Fédéré est le suivant : chaque entité participante (appelée *client*) entraîne un modèle sur ses propres données, puis transmet uniquement les paramètres appris (appelés *poids*) à un serveur central. Ce serveur agrège ces contributions pour mettre à jour un modèle global, qui est ensuite renvoyé aux clients. Ce processus est répété plusieurs fois jusqu'à convergence du modèle.

Plusieurs variantes méthodologiques existent dans la littérature. La plus courante est la méthode **FedAvg**, à laquelle s'ajoutent des extensions telles que **FedProx** (proximité) et **FedOpt** (optimisation adaptative). Toutes reposent sur deux grandes étapes :

— **Entraînement local** : chaque client cherche à minimiser sa fonction de coût locale $F_k(w_k)$, où k désigne le client. Pour FedAvg et FedOpt, il s'agit de la log-vraisemblance classique issue d'un modèle de régression logistique, adaptée à une variable cible binaire (sinistre ou non) supposée suivre une loi de Bernoulli. En FedProx, un terme de régularisation supplémentaire est ajouté à cette fonction de coût afin de limiter l'écart entre les poids locaux w_k et le modèle global w^{global}, stabilisant ainsi l'apprentissage dans des environnements hétérogènes.

— **Agrégation globale** : une fois l'entraînement local terminé, les poids w_k sont envoyés au serveur central qui les combine. La méthode FedAvg effectue une moyenne pondérée des poids, en tenant compte du nombre d'observations n_k de chaque client : $w^{\text{FedAvg}} = \sum_{k=1}^{K} \frac{n_k}{\sum_{j=1}^{K} n_j} w_k$

En alternative, la méthode FedOpt applique une stratégie d'optimisation plus complexe sur le serveur, intégrant des techniques similaires comme Adam, Adagrad et Yogi.

Soit E le nombre d'itérations locales effectuées avant communication avec le serveur, et n_{round} le nombre total de cycles d'entraînement. Le modèle local est mis à jour par :

$$w \leftarrow w - \alpha \nabla F_k(w; B) \quad \text{FadAvg et FedOpt}$$

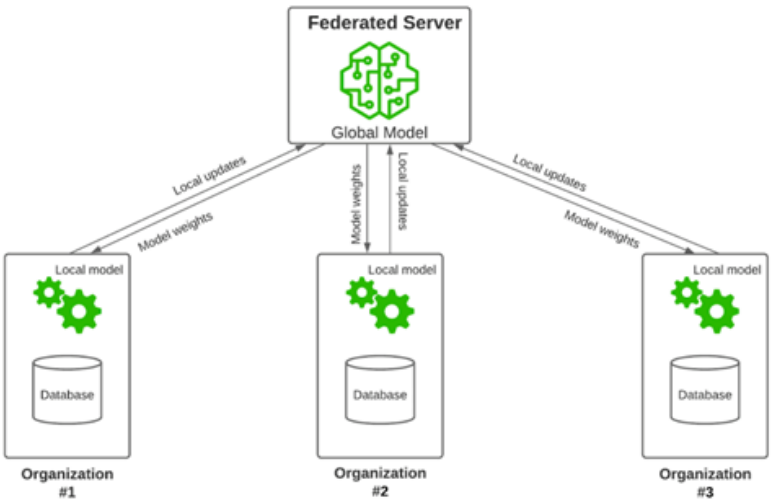

FIGURE 1 – Schéma fonctionnel du mécanisme de FL

où B est un mini-batch de données locales, α le taux d'apprentissage, et F_k la fonction de perte locale du client k. En effet, chaque données d'entraînement locale est divisée en mini batch. À chaque nouveau round, nous changeons de mini-batch, le nombre total de mini-batch étant égale au nombre de round totale.

Enfin, dans le cadre d'un projet assurantiel, les modèles de régression logistique sont largement privilégiés en raison de leur simplicité d'utilisation et de leur grande interprétabilité, qualités qui sont moins évidentes pour les modèles de réseaux de neurones. Chaque modèle local sera donc une régression logistique.

L'objectif est de prédire la probabilité de survenance d'un sinistre automobile, notée $P(y_i^k = 1 \mid x_i^k, w_k)$. Chaque observation est également associée à une durée d'exposition au risque sur une période d'une année, représentée par la variable `Exposure` $\in [0,1]$. Ainsi, notre probabilité de réaliser un sinistre sera pondérée à son `Exposure` : $P_{\text{réel}}(y_j^k = 1 \mid x_j^k, w_k) = P(y_j^k = 1 \mid x_j^k, w_k) \cdot \texttt{Exposure}_j$

3 Données

Avant d'entamer toute phase de modélisation, il est essentiel de disposer de jeux de données harmonisés et adaptés à l'apprentissage fédéré. Dans notre contexte, les clients sont représentés par trois assureurs, chacun disposant de données issues d'un pays différent : la France (freMTPL), la Belgique (beMTPL) et un ensemble européen (euMTPL). Ces bases, bien que similaires dans leur structure générale (exposition, sinistres, informations conducteur et véhicule), présentent des tailles et des distributions très hétérogènes comme le présente le tableau ci-dessous.

Base	Nombre d'observations
freMTPL - France	1 091 182
beMTPL - Belgique	163 212
euMTPL - Europe/Italie	2 373 197

TABLE 1 – Quantité de données par base

Un important travail de nettoyage, d'uniformisation et d'harmonisation des variables a été mené en amont. Certaines variables sont naturellement comparables (âge du conducteur, exposition annuelle), tandis que d'autres ont dû être normalisées, uniformisées (Fuel_type) ou reconstituées, comme Density ou Gender (absente dans la base française). Toutes les variables numériques ont été standardisées avec MinMaxScaler afin de faciliter l'optimisation au sein de chaque client.

Les résultats de la statistique descriptive ont mis en évidence un fort déséquilibre de la variable **Sinistre**, définie comme une variable binaire indiquant la survenue ou non d'un sinistre au cours de l'année. Ce déséquilibre varie selon les bases, avec un taux de sinistralité allant de 4,5% à 11,2%. Afin d'éviter que les modèles locaux ne soient biaisés par une prédominance des non-sinistres, nous avons eu recours à une stratégie de data augmentation, consistant à rééchantillonner les sinistres par interpolation contrôlée ou aléatoirement pour les variables categorielles. Cette technique a permis de rééquilibrer les classes dans les données d'entraînements tout en préservant les caractéristiques statistiques des autres variables. En effet, les distributions des variables explicatives sont restées très proches avant et après augmentation.

Enfin, une régression logistique classique a été appliquée localement, pour évaluer l'effet de chaque variable sur la probabilité de sinistre. Tous les coefficients estimés se sont révélés significatifs au seuil de 1% confirmant la pertinence des variables retenues et leur inclusion dans le modèle fédéré.

4 Implémentation et résultats

4.1 FedAvg

L'implémentation de l'algorithme FedAvg repose sur une série de régressions logistiques utilisant `SGDClassifier` avec mini-batchs, une régularisation $\mathscr{L}^2$ et une gestion du déséquilibre des classes. Chaque client (base) entraîne son modèle sur 60% de ses données augmentée et l'évalue sur les 40 % restantes, avant agrégation. Pour évaluer les performances du modèle fédéré, des comparaisons sont effectuées avec XGBoost, servant de borne supérieure théorique, et avec des régressions logistiques locales comme le montre le tableau ci-dessous.

Base	AUC XGBoost	AUC Régression Logistique
freMTPL - France	0.6729	0.55
beMTPL - Belgique	0.5987	0.56
euMTPL - Europe/Italie	0.6605	0.57

TABLE 2 – Comparaison des AUC obtenus avec XGBoost et régression logistique locale

L'algorithme FedAvg montre des performances globalement modestes, avec une AUC stabilisée autour de 0,619. Ce score, cohérent avec ceux des modèles locaux, reste bien en dessous de celui obtenu avec XGBoost, ce qui souligne les limites du modèle dans un contexte de données hétérogènes. L'instabilité est particulièrement marquée pour la base belge, dont la faible taille entraîne des mini-batchs peu fiables et des poids locaux erratiques, dégradant l'apprentissage global. À l'inverse, la base européenne, beaucoup plus fournie, domine largement l'agrégation, tirant le modèle vers ses propres coefficients. Cela crée un déséquilibre entre les clients. Si les coefficients globaux sont plus stables, ils reflètent surtout la dynamique du client le plus volumineux, ce qui limite l'adaptation aux spécificités locales. Au final, bien que FedAvg permette une certaine régularisation des poids, ses performances restent en retrait face aux modèles non fédérés, en particulier dans des environnements marqués par une forte hétérogénéité. Ces limites justifient le recours à FedProx et FedOpt dans la suite de la modélisation.

4.2 FedProx et FedOpt

FedProx, bien qu'introduisant un terme de régularisation censé stabiliser les mises à jour locales ($\mu = 0.1$), ne montre pas d'amélioration notable par rapport à FedAvg. Le score AUC global reste quasiment identique (0.618 contre 0.619), et les problèmes de la base belge persistent : performance faible (AUC < 0.55) et forte instabilité comme on peut le constater sur les graphiques suivants.

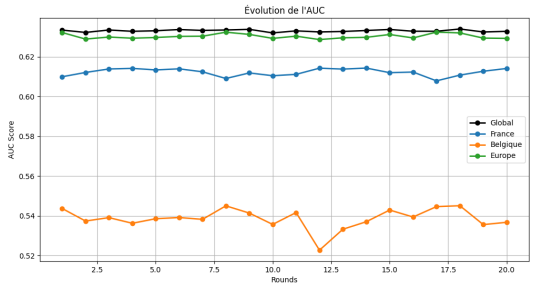

FIGURE 2 – Évolution du score AUC au fil des rounds pour FedProx, par région

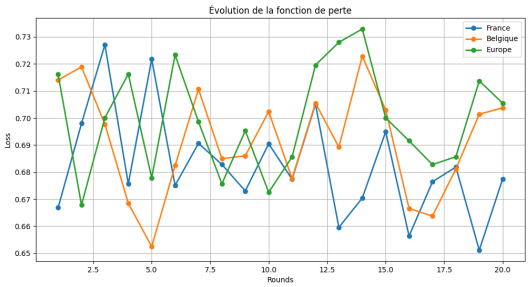

FIGURE 3 – Évolution de la fonction de perte (loss) au fil des rounds pour Fedprox

En revanche, FedOpt se distingue par une amélioration plus marquée, notamment sur la base européenne (AUC jusqu'à 0.6319), et une progression relative pour la Belgique (+3,92 % par rapport au modèle local). L'utilisation d'un optimiseur adaptatif (FedAdam) permet un apprentissage plus stable, avec des courbes de perte plus lissées et une convergence plus fluide. L'évolution des coefficients sous FedOpt montre une dynamique amortie, traduisant une meilleure stabilité dans l'agrégation. Certaines variables, comme Fuel_type ou DriverAge, évoluent de manière plus cohérente avec les effets locaux comme le présente la figure ci-dessous.

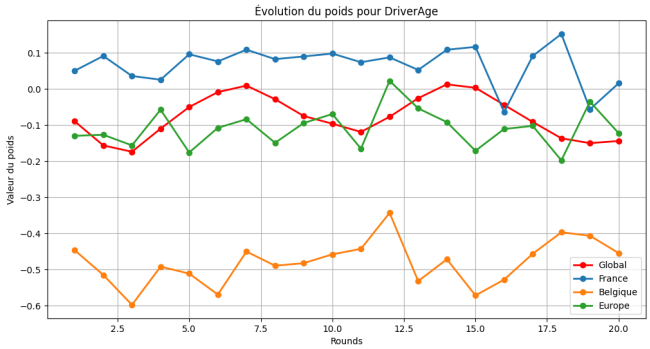

FIGURE 4 – Évolution des poids DriverAge par FedOpt (Adam)

Contrairement à FedAvg, FedOpt limite les fluctuations brutales des poids et atténue les biais régionaux. Enfin, même si la base européenne influence encore fortement le modèle global, FedOpt gère mieux l'hétérogénéité inter-clients et améliore la prise en compte des bases plus petites comme celle de la Belgique.

5 Conclusion

Ce projet a permis d'explorer l'apprentissage fédéré dans le secteur de l'assurance. Malgré des résultats AUC globaux en deçà des attentes, l'étude souligne l'influence des algorithmes d'agrégation sur la performance et l'interprétabilité. FedOpt, plus stable, s'est montré efficace, mais moins transparent que FedAvg ou FedProx. Les limites rencontrées sont principalement liées à la taille réduite et à l'hétérogénéité des bases, en particulier pour la Belgique. Cela a limité le nombre de rounds d'entraînement et entraîné une possible perte d'information locale lors de l'agrégation. Enfin, l'usage de SMOTENC sur les jeux d'entraînement a permis un rééquilibrage des classes, bien qu'une alternative comme l'under-sampling contrôlé aurait pu mieux préserver la nature des données. En résumé, l'apprentissage fédéré a du potentiel, mais il faut encore affiner les approches pour qu'il soit vraiment pertinent dans un contexte assuranciel.