

Mémoire présenté devant l'ENSAE Paris
pour l'obtention du diplôme de la filière Actuariat
et l'admission à l'Institut des Actuaires
le 05/02/2020

Par : **Thierry-Séphine GOMA**

Titre : **Apport des réseaux neuronaux au provisionnement IARD :
Application d'architectures récurrentes à la prédition
d'évolution temporelle de montants**

Confidentialité : NON OUI (Durée : 1 an 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de la filière
Christian Yann ROBERT

Entreprise : Mazars Actuariat 
Nom : Alexandre GUCHET
Signature :

*Membres présents du jury de l'Institut
des Actuaires*
Emmanuel DUBREUIL
Florence PICARD
Adrien SURU
Marc JUILLARD

Directeurs du mémoire en entreprise :
Noms : Axel TRUY, Florian MONTANIER
Signatures :

*Autorisation de publication et de
mise en ligne sur un site de
diffusion de documents actuariels
(après expiration de l'éventuel délai de
confidentialité)*
Signature du responsable entreprise

Secrétariat :



Signature du candidat

Bibliothèque :



Résumé

L'activité d'assurance est caractérisée par la très forte incertitude quant aux coûts de revient des sinistres, et l'évaluation des provisions techniques constitue un enjeu stratégique pour la pérennité de l'assureur. Dans ce contexte la recherche de nouveaux modèles de provisionnement, en vue d'améliorer l'évaluation du montant des prestations futures versées par l'assureur, est un sujet d'intérêt. Parmi les nouvelles approches présentées, plusieurs s'inscrivent dans le déploiement de l'apprentissage profond pour la résolution de problématiques assurantielles, et proposent des méthodes de provisionnement s'appuyant sur les réseaux de neurones. Elles ont motivé le présent mémoire.

Nous nous sommes intéressée à l'apport des réseaux de neurones au provisionnement IARD et avons proposé l'implémentation d'un modèle de provisionnement par réseaux neuronaux récurrents, qui mobilisent des structures adaptées aux données séquentielles et à même d'apprendre des dépendances de long terme. Le modèle implémenté se propose de prédire conjointement les montants de paiements et de charges.

Afin d'en mesurer les performances, nous avons procédé à une étude comparative entre notre modèle et une méthode de référence : Mack-Chain-Ladder. Nous avons construit une mesure de performance et mené une étude approfondie des distributions des erreurs. Enfin, notre étude s'est intéressée à la compréhension des résultats. A cet effet nous avons mis en évidence les spécificités de notre modèle, procédé à des analyses de sensibilité du réseau, et finalement fait usage d'un outil d'aide à l'interprétation qui se propose d'apporter une compréhension locale de choix réalisés au cours d'une prédiction.

Mot clefs : provisionnement, apprentissage profond, deep learning, réseau de neurones recurrent (RNN), GRU, IARD, PSAP.

Abstract

The insurance business is characterized by a very high uncertainty regarding the cost of claims, and the valuation of technical provisions is a strategic issue for the insurer's long-term survival. In this context, the research for new reserving models, with a view to improve the evaluation future costs, is a subject of interest. Among the new approaches presented, several fall under the deployment of deep learning in the resolution of insurance problems, and propose reserving methods based on neural networks. They have motivated this thesis.

We studied the inputs and contribution of neural networks to P&C reserving and we have offered the implementation of a reserving model using recurrent neural networks, which are fitted to sequential data and have the ability to learn long-term dependencies. The model we have implemented offers to jointly predict the amounts of paid claims and case reserves.

To measure its performance, we have carried out a comparison between our model and a reference method : Mack-Chain-Ladder. We have built a performance metric and conducted an in-depth study of the error distributions. Finally, our study focused on understanding the results. To that end, we highlighted the specificities of our model, performed sensitivity analysis of the network, and finally resorted to an interpretation framework aiming to provide a local understanding of the choices made during a prediction.

Keywords : reserving, deep learning, recurrent neural network (RNN), GRU, P & C.

Note de synthèse

Contexte

L'activité d'assurance est caractérisée par une très forte incertitude quant au coût de revient des engagements pris par l'assureur. Dans ce cadre, le provisionnement est un enjeu stratégique pour l'assureur, qui cherche à évaluer les montants afférents au règlement des sinistres qu'il a pris l'engagement d'assurer. Confronté aux limites des méthodes de provisionnement classiques, l'emploi de l'apprentissage profond à des fins de résolution de problématiques actuarielles a connu un essor prononcé au cours des trois dernières années. Dans ce contexte, nous avons consacré notre mémoire à l'étude de l'apport des réseaux neuronaux au provisionnement IARD. Une première phase de revue de l'état de l'art nous a conduit à nous intéresser à une architecture spécifique munie de structures récurrentes : *Deep Triangle*, (KUO, 2018).

Présentation du modèle

Particularités

Le modèle *Deep Triangle* que nous avons sélectionné pour notre étude se distingue par trois aspects.

- Tout d'abord il se propose de prédire conjointement les montants de paiement et les montants de réserve dossier/dossier. A l'instar de la méthode Munich-Chain-Ladder, cette approche permet de prendre en compte des dynamiques qui ne le sont pas dans des modèles plus triviaux.
- La deuxième spécificité du modèle est qu'il propose de prédire les montants futurs en prenant en compte pour chaque année de survenance, la totalité de l'historique de développement. Ce fonctionnement pourrait prendre en compte des *patterns* de développement non considérés par Chain-Ladder.
- Enfin il met en œuvre des techniques de prédiction basées sur le *Deep Learning*, qui se sont illustrées par des résultats probants dans la résolution d'autres problèmes de prédiction de données séquentielles.

Architecture du réseau de neurones

L'architecture de *Deep Triangle* intègre deux éléments particuliers. Le premier est une unité récurrente *GRU* ou *Gated Recurrent Unit*, qui permet au réseau de neurones d'être doté de propriétés de mémoire et d'apprendre des dépendances de long terme. Le deuxième élément est une couche *embedding*. Elle permet de convertir les identifiants des compagnies en vecteurs dont la distance au sens euclidien est porteuse d'information sur les similarités entre les entreprises qu'ils représentent. Les propriétés de ces deux éléments leurs confèrent un potentiel intéressant dans des perspectives d'automatisation du provisionnement et de réduction du temps d'analyse actuarielle.

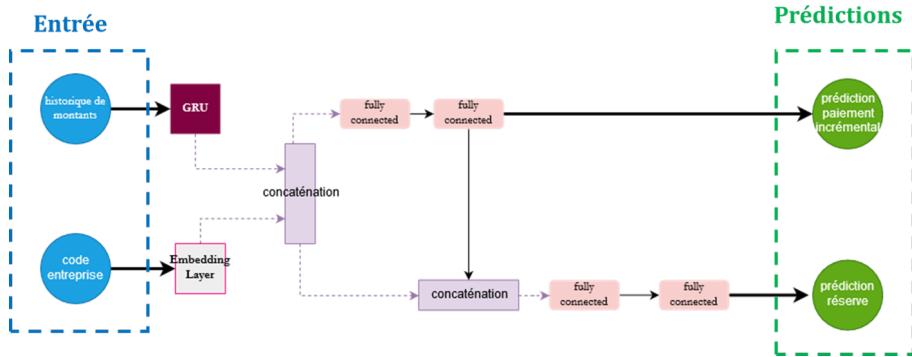


FIGURE 1 – Architecture du réseau *Deep Triangle*

Construction du modèle

Nous avons implémenté et entraîné notre modèle en langage Python en utilisant la librairie *Keras* (CHOLLET et al., 2015). Nous avons utilisé des données réelles, provenant de la base de données *Schédule P* de la *NAIC*, organisme américain de normalisation et d'appui à la réglementation assurantielle. Notre base de données comporte les historiques de déroulement d'agrégats de sinistres pour 4 secteurs d'activité différents.

Performances du modèle

Les performances de notre modèle ont été évaluées par une approche comparative avec une méthode de provisionnement de référence : Mack-Chain-Ladder. Cette dernière produisant des résultats pouvant faire l'objet de retraitements par les actuaires, nous avons implanté une option d'inclusion de retraitements des valeurs aberrantes. Nous avons mesuré les performances de deux façons : d'une part en construisant une mesure quantitative synthétisant la performance, et d'autre part en menant une étude de la distribution globale et une analyse des queues de distribution.

Pour le premier aspect de l'évaluation, nous avons construit une métrique adaptée à l'intérêt métier. Nous nous sommes intéressée aux écarts de prédition rapportés au volume de primes perçues. Pour chaque agrégat de sinistres C_i , on définit ainsi l'erreur de prédition à horizon J :

$$\delta_{C_i}^J = \frac{\hat{C}_{i,J} - C_{i,J}}{NEP_i}$$

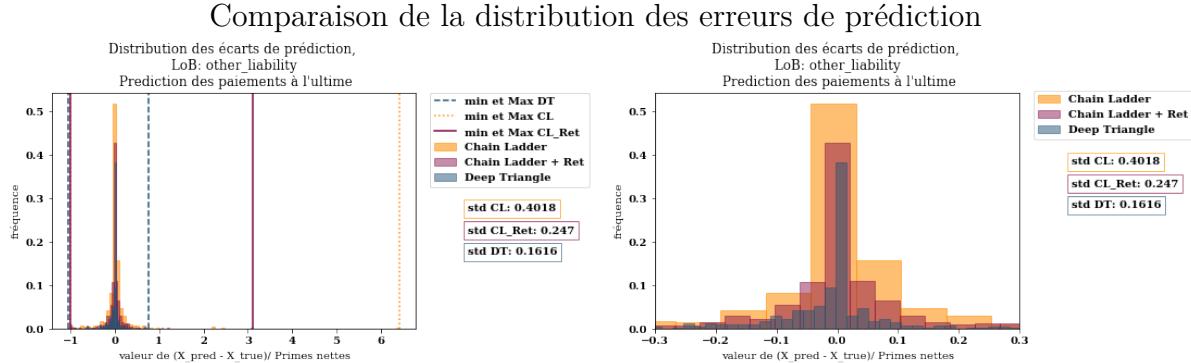
Par le calcul de ces mesures pour tous les agrégats de sinistres de notre base de données, une distribution des erreurs de prédition est obtenue. Une statistique de cette distribution nous a permis de construire une mesure quantitative synthétisant la performance : nous avons retenu l'erreur quadratique moyenne. Les résultats de cette première mesure sont les suivants :

TABLE 1 – Comparaison des prédictions de notre modèle avec la méthode de Chain-Ladder ainsi qu’avec une approche de type Chain-Ladder incluant des retraitements des valeurs aberrantes.

Erreurs quadratiques moyennes, <i>RMSE</i> pour la prédiction des paiements			
<i>LoB</i>	<i>DT</i>	CL	CL_Ret
<i>com</i>	0,1186	0,1472	0,1438
<i>oth</i>	0,1639	0,4027	0,2469
<i>pri</i>	0,0518	0,0725	0,0678
<i>wor</i>	0,1573	0,1743	0,1700

Cette mesure a permis de mettre en évidence les résultats probants de notre modèle pour la prédiction de paiements, et sa capacité à surpasser Chain-Ladder sur des *LoBs* longues. Elle souligne également l’amélioration des résultats de Chain-Ladder, induite par l’intégration de retraitements.

Dans un deuxième temps, nous avons comparé les distributions d’erreurs obtenues pour chaque modèle : nous avons étudié leurs supports, leur superposition et les erreurs dépassant un seuil de tolérance, déterminé par une approche actuarielle.



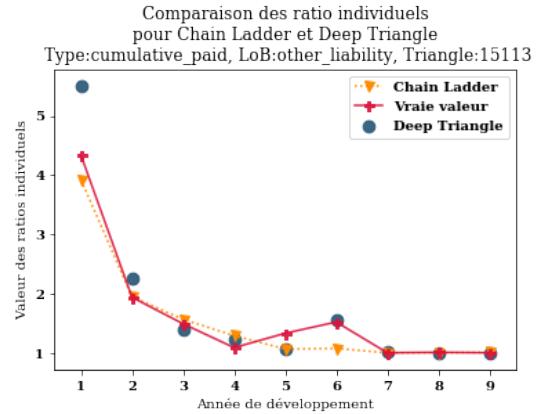
Cette étude a permis de mettre en évidence la stabilité du modèle *Deep Triangle* en comparaison de Chain-Ladder, qui conduit davantage à la prédiction de valeurs extrêmes, y compris lorsque des retraitements des valeurs aberrantes sont effectués.

Étude des comportements spécifiques au modèle : mise en évidence et tentative de compréhension

Des études nous ont permis de mettre en évidence plusieurs caractéristiques par lesquelles notre modèle se démarque de la méthode de Chain-Ladder.

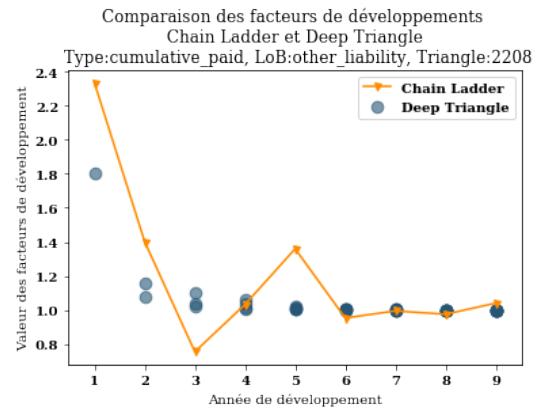
Articulation des deux historiques pour la prédiction

Notre modèle exhibe une capacité à exploiter conjointement les historiques de charges et de paiements reçus en entrée dans la prédiction d'un montant de paiements ou d'un montant de charges. Nous présentons sur le graphe ci-contre, comment cette aptitude à prendre en compte les charges permet de déceler des évolutions surprenantes des paiements, là où Chain-Ladder commet une erreur de prédiction importante.



Capacité de lissage de la donnée

Un comportement saillant du modèle est la capacité de lissage qu'il présente, traduisant un apprentissage du traitement automatique des valeurs aberrantes. Dans plusieurs cas, les prédictions de *Deep Triangle* se rapprochent des retraitements qui seraient réalisés par un actuaire. C'est cette caractéristique qui nous a conduit à intégrer à Chain-Ladder une heuristique de retraitement des valeurs aberrantes en vue de comparer les performances des deux modèles.



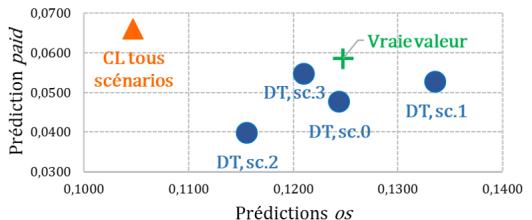
Sensibilité à la dynamique des paiements

Dans une étude d'approfondissement, nous avons entrepris de confirmer la prise en compte par notre modèle de la dynamique du déroulement des sinistres. Pour ce faire, nous avons construit différents scénarios d'historiques conservant le montant de paiement cumulés. Chaque scénario conduit à une prédiction différente de *Deep Triangle* :

Mise en évidence de l'impact de la dynamique du déroulement des sinistres. $\epsilon = 0.2$.

Historique	incrément $j - 3$	incrément $j - 2$	incrément $j - 1$
fidèle	0,1630	0,2205	0,1107
perturbé(1)	0,1630+ ϵ	0,2205- ϵ	0,1107
perturbé(2)	0,1630- ϵ	0,2205	0,1107+ ϵ
perturbé(3)	0,1630	0,2205+ ϵ	0,1107- ϵ

Résultats de prédiction pour les différents scénarios, $\epsilon=0.2$



Ce résultat marque une autre différence majeure avec la méthode de Chain-Ladder qui s'appuie sur des montants cumulés et conduit donc à la même prédiction dans chacun des scénarios.

Recours à un outil d'aide à l'interprétabilité

La capacité à interpréter les choix de notre modèle a été une tâche à laquelle nous nous sommes intéressée. Après une première étude au cours de laquelle nous avons perturbé marginalement chacun des montants historiques en vue de mesurer l'évolution des prédictions en sortie, nous avons employé un outil d'aide à l'interprétabilité en accompagnement de nos prédictions, le *framework LIME* (RIBEIRO et al., 2016b). Cet outil se propose d'expliquer localement les prédictions de tout modèle et il nous a permis d'identifier les variables explicatives impactant significativement une prédition, ainsi que d'obtenir des conditions déterminant dans quelles mesures ces variables influent sur la prédition de montants.

Limits

Les capacités d'interprétation locale que nous avons acquises grâce à l'outil *LIME*, nous permettent de mettre en évidence pour un historique, les variables les plus importantes pour la prédition ainsi que de quantifier précisément les seuils qui conduisent ces variables à peser sur l'augmentation ou la diminution de la prédition future. Une lecture des hypothèses peut être faite, il ne s'agit toutefois que d'une explication possible, et qui est propre à chaque prédition. Une piste pertinente serait d'acquérir la capacité de traduire ces hypothèses en termes de propriétés assurantielles des sinistres et à une échelle macro. Ainsi, il peut sembler trop tôt pour pouvoir utiliser notre modèle comme modèle de provisionnement, compte tenu des exigences du régulateur dans la compréhension des modèles.

Perspectives

Le modèle *Deep Triangle* ouvre des perspectives d'automatisation et pourrait, par exemple, être utile dans le cadre de processus de revue indépendante de provisionnement, en tant que modèle *challenger*. Il permettrait une identification rapide des cas nécessitant une revue détaillée. Le modèle ouvre également des perspectives d'élargissement de la modélisation des graves, pour lesquels on dispose d'un historique de charges et de développement du sinistre et dont on cherche à améliorer l'estimation de la charge ultime. Enfin, du fait du caractère spécifique des données employées pour calibrer notre modèle, il serait pertinent de mettre à l'épreuve l'architecture *Deep Triangle* avec des données d'assureurs français. On pourrait notamment choisir de se concentrer sur des lignes à risque telles que la responsabilité civile médicale ou la responsabilité civile construction.

Executive summary

Context

The insurance business is characterized by a very high degree of uncertainty as to the future cost of the insurer's engagements. In this context, reserving is a strategic issue for the insurer, seeking to evaluate the amounts pertaining to the settlement of future claims. Faced with the limitations of traditional reserving methods, the use of Deep learning to tackle actuarial problems has grown significantly over the last three years. In this context, we have dedicated our thesis to studying the inputs and contribution of neural networks to P&C reserving. A preliminary stage reviewing the state of the art has led us to focus on a specific architecture which encompasses recurrent structures : *Deep Triangle*, (KUO, 2018).

Model presentation

Characteristics

The model *Deep Triangle* we have selected to conduct our study is distinctive in three ways.

- First of all, the model permits to predict jointly the amounts for paid claims and case reserves. Such as Munich-Chain-Ladder method, this approach allows to consider dynamics that are not taken into account by more trivial models.
- The second specificity of the model is that it uses the entire development history to perform prediction of future amounts. This could lead to taking into account development patterns that are not considered by Chain-Ladder method.
- Finally, it applies predictive techniques based on Deep Learning, which have proven successful in solving other sequential data prediction problems.

Neural network architecture

Deep Triangle's architecture incorporates two particular elements. The first one is a Gated Recurrent Unit, GRU, which provides memory ability to the neural networks and enables it to learn long-term dependencies. The second element is an embedding layer. This layer proceeds to the conversion of company identifiers into multidimensional vectors, in such way that the Euclidean distance between two vectors carries information and reflects the similarities between the companies. These properties provide these elements with an interesting potential in automation perspectives, with a view to reduce actuarial analysis time.

Implementation

We implemented and trained our model in Python language, using *Keras* (CHOLLET et al., 2015) library. We used real data from the *Schedule P* database of the American National Association of Insurance Commissioners (NAIC). Our database contains historical claim development data from 4 different lines of business.

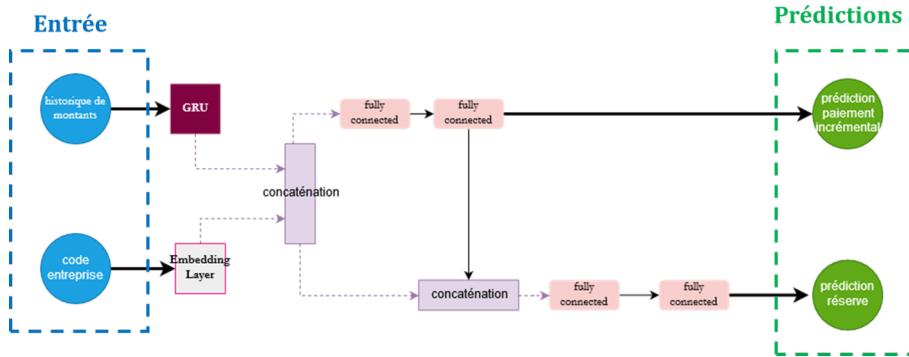


FIGURE 2 – Deep Triangle network architecture

Performance

The performance of our model has been assessed using a comparative approach with a reference reserving method : Mack-Chain-Ladder. As the latter produces results that sometimes need to be adjusted by actuaries, we implemented an option to include reprocessing of the outliers. We measured performance of the models according to two approaches : first by creating a quantitative measure which summarizes performance, and secondly by conducting a study of the global distribution followed by a tail analysis.

For the first approach, we built a metric adapted to the business interest. We considered the prediction deviations relative to the amount of premium earned. For each claim aggregate C_i , the prediction error at horizon J is defined as :

$$\delta_{C_i}^J = \frac{\hat{C}_{i,J} - C_{i,J}}{NEP_i}$$

By calculating these measures for all paid claims in our database, a distribution of prediction errors is obtained. A statistic of this distribution allowed us to construct a quantitative measure of performance : we have used the root mean square error. The results of this first measurement are as follows :

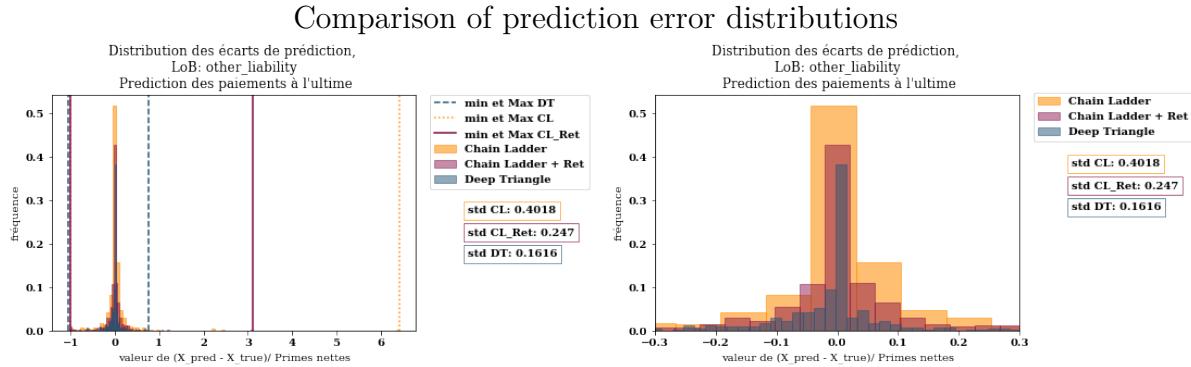
TABLE 2 – Comparison of prediction performance including our model, the Chain-Ladder method and a Chain-Ladder approach performing outlier reprocessing.

LoB	Root mean square error for the prediction of the paid claims		
	DT	CL	CL_Ret
com	0.1186	0.1472	0.1438
oth	0.1639	0.4027	0.2469
pri	0.0518	0.0725	0.0678
wor	0.1573	0.1743	0.1700

This measure has enabled us to highlight the convincing results of our model for the prediction of future paid claims, as well as its ability to outperform Chain-Ladder over

long *LoBs*. It also underlined the improvement brought by the integration of reprocessing in Chain-Ladder results.

Secondly, we compared the error distributions of each model : we studied their supports, their superposition and the errors exceeding a level of tolerance for which the threshold has been determined by an actuarial approach.



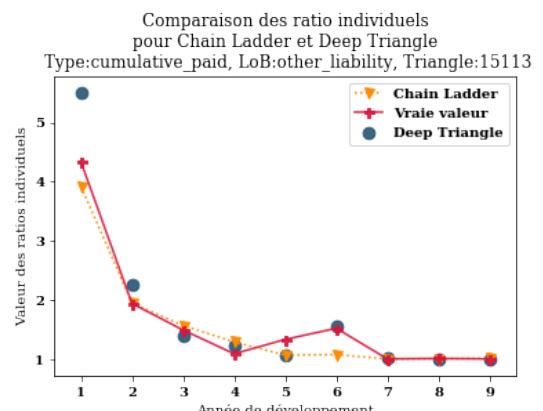
This study highlighted the stability of the *Deep Triangle* model compared to Chain-Ladder, which is more prone to lead to the prediction of extreme values, even when performing outlier reprocessing.

Study of model-specific behaviours : highlight of particularities and attempt to interpretation

Studies have enabled us to highlight several characteristics by which our model differs from the Chain-Ladder method.

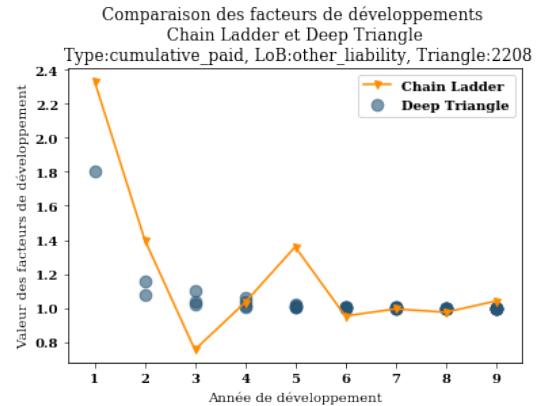
Articulation of both data type to predict

Our model exhibits an ability to jointly exploit the history of case reserve and paid claims to predict a claim amount or a case reserve amount. On the graph opposite, we show how this ability to take into account case reserves allows us to detect surprising trends in paid claims, while Chain-Ladder makes an important prediction error.



Smoothing ability

A prominent behavior of the model is the smoothing capability it exhibits, reflecting a learnt automatic processing of the *outliers*. In several cases, the predictions of *Deep Triangle* is close to the restatements that would be performed by an actuary. This characteristic led us to design a heuristic to reprocess outlier for Chain Ladder, in order to compare the performances of the two models.



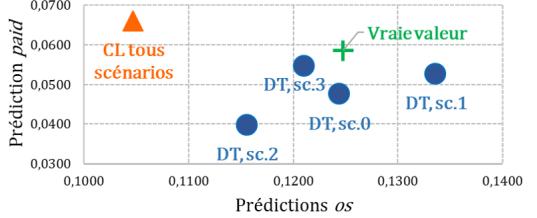
Sensitivity to development dynamics

Furthering our study, we undertook to confirm that our model takes into account the dynamics of the claims development. To do so, we built different scenarios keeping the cumulative amount of paid claim unchanged. Each scenario leads to a different prediction by *Deep Triangle* :

Highlight of the impact of claims development dynamics. $\epsilon = 0.2$.

History	incremental $j - 3$	incremental $j - 2$	incremental $j - 1$
faithfull	0,1630	0,2205	0,1107
disturbed(1)	0,1630+ ϵ	0,2205- ϵ	0,1107
disturbed(2)	0,1630- ϵ	0,2205	0,1107+ ϵ
disturbed(3)	0,1630	0,2205+ ϵ	0,1107- ϵ

Résultats de prédiction pour les différents scénarios, $\epsilon=0.2$



This result constituted another major difference with the Chain-Ladder method, which relies on cumulative amounts and therefore leads to the same prediction in each of the scenarios.

Using an interpretability framework

The ability to interpret the choices made by our model is an aspect we have worked on. Following an initial study in which we marginally disrupted each of the historical amounts in order to measure the evolution of the output predictions, we used an interpretability framework to conduct our predictions, the framework LIME (RIBEIRO et al., 2016b). This tool is intended to explain the predictions of any model locally and it has allowed us to identify the explanatory variables that significantly impact a prediction, as well as to obtain conditions that rule in which way these variables influence the prediction.

Limits

The local interpretation capacities that we have acquired thanks to the LIME tool, allowed us to highlight the most important variables for a prediction, given a historic,

as well as to quantify precisely the thresholds leading these variables to weigh on the increase or decrease of the future prediction. An interpretation of the hypotheses can be made, however it is only hypothetical, and specific to each prediction. A relevant pursuit would be to acquire the ability to translate these assumptions on a macro scale and into insurance properties for the claims. Thus, it may seem too early to use our model as a provisioning model, given the regulator's requirements in understanding models.

Perspectives

Deep Triangle model opens up perspectives of automation and could, for example, be useful in the context of an independent reserving review, as a *challenger* model. It would enable a reliable and time-effective identification of the lines of business requiring a more thorough review. The model also opens up prospects for expanding the modelling of serious claims, as historical data is available and could be put to use when seeking to improve the prediction of the final cost. Finally, given the specificity of the data used to train our model, it would be interesting to test the architecture of *Deep Triangle* with data from French insurers. In particular, we could choose to focus on highly risky lines of business such as medical liability or construction liability.

Remerciements

Je tiens à remercier toutes les personnes au contact desquelles j'ai été amenée à travailler au cours des mois passés chez Mazars. Leur disponibilité et leur attention ont contribué à la réalisation de mon mémoire dans des conditions favorables.

Nommément je remercie Alexandre GUCHET et Grégory BOUTIER, pour l'intérêt qu'ils accordent à la réalisation des mémoires chez Mazars Actuariat. Je remercie Axel TRUY pour son encadrement rigoureux et bienveillant et pour son expertise éclairante. Je remercie Florian MONTANIER pour son encadrement consciencieux et pour ses conseils précis.

Je veux également remercier chacun des membres de l'équipe pédagogique de l'ENSAE Paris qui m'ont accompagnée, enseignée, et conseillée au cours des quatre dernières années.

Enfin, je remercie toutes les personnes qui m'entourent dont le soutien m'a été précieux.

Sommaire

Résumé	ii
Abstract	ii
Note de synthèse	iii
Executive summary	viii
Remerciements	xiii
Introduction : La résolution de problématiques assurantielles par le <i>Deep Learning</i>	1
I Une présentation du provisionnement en assurance IARD et des réseaux de neurones	3
1 Nécessité et caractérisation du provisionnement en assurance IARD	5
1.1 Pourquoi constituer des provisions techniques en assurance?	5
1.1.1 La notion d'aléa inhérente à l'activité assurantielle	5
1.1.2 L'enjeu des provisions techniques en assurance IARD	5
1.1.3 Les provisions techniques dans le cadre de notre étude	6
1.1.3.1 La provision pour sinistre à payer	6
1.1.3.2 Déroulement et gestion d'un sinistre	7
1.2 Les multiples approches de provisionnement selon les différentes normes . .	9
1.2.1 Comptes sociaux : évaluation prudente	9
1.2.2 Solvabilité 2 : évaluation économique <i>Best Estimate</i>	10
1.2.3 IFRS17 : vision consolidée des comptes	11
2 La réalisation du provisionnement par les actuaires	13
2.1 Un modèle classique de provisionnement : Mack-Chain-Ladder	13
2.1.1 Formalisation préliminaire : représentation d'un triangle de sinistres	13
2.1.2 Mise en œuvre du modèle de Mack-Chain-Ladder	15
2.1.2.1 Description du modèle	15
2.1.2.2 Application sur un exemple	16
2.1.3 Limites et remise en cause de Chain-Ladder	17
2.2 L'essor de méthodes de provisionnement reposant sur le <i>Deep Learning</i> . .	18
2.2.1 Un intérêt croissant	18
2.2.2 Des publications marquées au cours des trois dernières années . .	18

3 Les réseaux de neurones	21
3.1 Présentation du neurone unique : formalisation mathématique	21
3.2 Entraînement du neurone unique pour une tâche de prédiction	23
3.2.1 L'échantillon d'entraînement	24
3.2.2 La fonction de perte et la fonction de coût	24
3.2.3 L'algorithme d'optimisation : la descente de gradient	25
3.3 L'utilisation des réseaux de neurones	26
3.3.1 Les limites du neurone unique	26
3.3.2 Présentation des réseaux de neurones	27
3.3.3 Capacité d'estimation des réseaux de neurones	28
3.4 Les réseaux de neurones récurrents	29
3.4.1 Caractérisation des réseaux de neurones récurrents	29
3.4.2 La mémoire des réseaux de neurones récurrents	30
3.4.3 <i>Le Gated Recurrent Unit</i>	31
3.4.3.1 Fonctionnement du <i>GRU</i>	31
3.4.3.2 Utilisation de l'unité récurrente à portes <i>GRU</i> dans un réseau de neurones	32
II Le modèle <i>Deep Triangle</i> de Kévin Kuo	35
4 Une présentation du modèle <i>Deep Triangle</i>	37
4.1 Présentation du problème à résoudre	38
4.1.1 Les variables explicatives utilisées	38
4.1.2 Une formalisation de la tâche d'apprentissage	39
4.2 L'architecture du réseau de neurones	40
4.2.1 Aperçu de l'architecture	40
4.2.2 Une couche <i>embedding</i>	40
4.2.3 Un bloc <i>GRU</i>	41
4.3 Paramètres d'entraînement du modèle	41
4.3.1 Fonction de coût	41
4.3.2 Algorithme d'optimisation <i>AMSGrad</i>	42
5 Une exploration des données utilisées	43
5.1 Une description statistique des données de la base d'étude	43
5.1.1 Dictionnaire des variables	43
5.1.2 L'activité des individus de notre base de données	44
5.1.3 Le profil et les flux financiers de nos entreprises	46
5.1.3.1 Les facteurs de développements sur chacune des <i>LoBs</i>	46
5.1.3.2 Les ratios de perte	50
5.1.3.3 Poids des entreprises dans chaque <i>LoB</i> selon leur volume de primes	51
6 Notre implémentation du modèle	53
6.1 Utilisation des données pour la construction du modèle	53
6.1.1 Constitution des échantillons d'entraînement et de test	53

6.1.2	Emploi de chacun des développements des sinistres	54
6.2	Description des différentes phases du processus d'entraînement du réseau de neurones	55
6.2.1	Étape 1 : détermination du nombre optimal d'époques	55
6.2.2	Etape 2 : détermination de la valeur optimale des poids du réseau	56
6.3	Utilisation du réseau de neurones entraîné pour la prédiction : conception d'un schéma itératif	58
III	Mesures de performances et approfondissement de l'étude du comportement du modèle	59
7	Mesures de performances du modèle	61
7.1	Utilisation d'une statistique synthétisant la performance et comparaison des performances de notre modèle avec une méthode de référence	61
7.2	Performances pour la prédiction à 1 an	62
7.3	Prédiction à l'ultime	65
7.3.1	Mesure quantitative synthétisant la performance	65
7.3.1.1	Calcul de la performance globale	65
7.3.1.2	Impact de la durée de l'historique d'information sur la qualité des prédictions	66
7.3.2	Étude de la distribution des erreurs commises	67
7.3.3	Étude des prédictions dépassant un seuil de tolérance donné	69
7.3.3.1	Analyse des prédictions extrêmes pour une <i>LoB</i> longue	69
7.3.3.2	Population des queues de distribution pour l'ensemble des <i>LoBs</i>	72
8	Mise en évidence du comportement du réseau de neurones	75
8.1	Multiplicité des facteurs de développement et articulation des informations	75
8.1.1	Des valeurs multiples pour les facteurs de développement en opposition à une valeur unique pour Chain-Ladder	75
8.1.2	Prise en compte du montant de charges	76
8.2	Capacité de lissage du modèle	77
8.2.1	Comparaison du traitement des <i>outliers</i> par <i>Deep Triangle</i> et par Chain-Ladder	77
8.2.2	Intégration de retraitements dans la prédiction par Chain-Ladder	78
8.2.2.1	Description de l'heuristique de retraitement des valeurs aberrantes	78
8.2.2.2	Résultats de l'intégration de retraitements	79
9	Étude et interprétation du comportement du modèle	83
9.1	Étude de la sensibilité du modèle à la couche <i>embedding</i>	83
9.1.1	Analyse de la spatialisation réalisée par la couche <i>embedding</i>	83
9.1.1.1	Analyse en composantes principales des vecteurs obtenus	83
9.1.1.2	Exploration de la proximité entre chacun des groupes d'assurance pour une <i>LoB</i>	84

9.1.2	Étude de la sensibilité du modèle à l'usage de la couche <i>embedding</i> .	85
9.2	Étude de l'impact de la dynamique de déroulement du sinistre dans la prédition	86
9.3	Sensibilité à l'introduction de perturbations dans les historiques de sinistres	88
9.4	Utilisation de <i>LIME</i> , outil d'aide à la compréhension en <i>Machine Learning</i>	91
9.4.1	Fonctionnement de <i>LIME</i>	91
9.4.2	Résultats d'explication d'une prédition	91
Conclusion		94
Références bibliographiques		97
A Complément de statistiques descriptives		103
A.1	Analyse des ratios de perte en distinguant les différents secteurs d'activité .	103
A.2	Analyse des facteurs de développement médians et leur variance sur chaque <i>LoB</i>	104
B Annexes aux résultats de mesure des performances		107
B.1	Mise en évidence de l'impact du <i>lag</i> initial sur l'erreur	107
B.2	Distributions empiriques des erreurs de prédition et résultats de l'inclusion de retraitements dans la prédition par Mack-Chain-Ladder	108
C Mise en évidence du comportement du modèle <i>Deep Triangle</i>		111
C.1	Courbes représentatives de l'impact marginal des données historiques sur la prédition	111

Introduction : La résolution de problématiques assurantielles par le *Deep Learning*

Le recours aux réseaux de neurones a montré des résultats probants dans la résolution de problématiques d'assurance au cours des dernières années. Parmi ceux-ci on peut notamment citer la détection de conduites dangereuses au volant, introduite en 2016 par l'assureur américain StateFarm, pouvant faire l'objet d'application en tarification, ou encore l'interface d'estimation à partir d'une image des coûts de réparation d'un véhicule automobile introduite en 2017 par la compagnie Liberty Mutual Insurance. Dans le même temps, les réseaux de neurones s'illustrent également dans les problématiques de prédiction de données séquentielles.

Les très bons résultats obtenus par leur application en assurance, ont encouragé les études s'intéressant à leur usage dans la résolution de problématiques éminemment actuarielles telles que la constitution de provisions. Cette tâche, qui consiste en l'évaluation de l'engagement de l'assureur, est absolument cruciale à l'activité d'assurance et est un enjeu majeur de par son poids dans le bilan. Cependant, les solutions de provisionnement existantes présentent des limites : reposant sur des hypothèses fortes, elles produisent des résultats qui nécessitent de faire l'objet de retraitements par des experts.

Dans ce contexte, nous nous proposons d'étudier l'apport des réseaux neuronaux dans la résolution de cette problématique de prédiction qui se présente comme un sujet d'intérêt et d'actualité.

Pour mener cette étude, nous nous intéresserons dans un premier temps à la tâche de provisionnement ainsi qu'aux outils envisagés pour la résolution de ce problème. Après avoir discuté du cadre de notre étude et mis en évidence l'essor des méthodes de provisionnement s'appuyant sur les réseaux neuronaux, nous offrirons une présentation du fonctionnement des réseaux neuronaux.

La deuxième partie de notre étude sera consacrée à l'analyse d'un modèle spécifique s'appuyant sur un type de réseau de neurones en particulier : les réseaux de neurones récurrents. Nous présenterons le modèle que nous avons implémenté, ainsi que les données que nous avons utilisées. Nous exposerons ensuite le processus d'entraînement par lequel nous avons procédé au calibrage du modèle.

Enfin nous nous attacherons dans une troisième partie à une étude approfondie des résultats fournis par le modèle. Dans un premier temps, nous procéderons à une mesure des performances de notre modèle. A cet effet, une comparaison avec une méthode de provisionnement de référence nous permettra une évaluation pertinente. Nous comparerons dans un premier temps les deux modèles en construisant une mesure synthétisant la

performance et adaptée à l’enjeu métier. Puis, nous considérerons les distributions des erreurs de chaque modèle, et étudierons la stabilité des modèles en analysant les prédictions extrêmes, au delà d’un seuil de tolérance que nous déterminerons. Nous nous intéresserons ensuite à une compréhension des résultats de notre modèle. Nous mettrons en évidence les comportements spécifiques du modèle, et analyserons la sensibilité de la prédiction à des perturbations introduites. Enfin nous aurons recours à un outil d’aide à l’interprétabilité en vue d’acquérir une capacité d’interprétation locale de nos prédictions de montants.

Première partie

**Une présentation du
provisionnement en assurance IARD
et des réseaux de neurones**

Chapitre 1

Nécessité et caractérisation du provisionnement en assurance IARD

1.1 Pourquoi constituer des provisions techniques en assurance ?

1.1.1 La notion d'aléa inhérente à l'activité assurantielle

La notion d'aléa et d'incertitude est inhérente à l'activité d'assurance. Un contrat d'assurance se présente comme un accord entre plusieurs parties, aux termes duquel l'assureur vend au souscripteur une protection contre la survenance d'un événement incertain. Cette protection prend la forme d'une prestation. Le contrat implique un assuré et un bénéficiaire, qui peuvent être des personnes différentes du souscripteur. Le contrat d'assurance prévoit ainsi qu'en cas de survenance d'un sinistre affectant l'assuré, l'assureur verse une prestation au bénéficiaire.

Le contrat d'assurance fournit donc une couverture contre les risques liés à la survenance d'un événement probable, en échange du versement d'une prime dont le montant est fixé à la souscription du contrat. Si la nature de la prestation est déterminée au moment de la souscription du contrat d'assurance, le montant de la prestation qui sera versé au bénéficiaire au terme du contrat n'est pas encore connu à la souscription du contrat, puisque qu'il est empreint d'incertitude.

Cette explication concise nous permet de mettre en évidence la caractéristique la plus essentielle de l'activité d'assurance : l'assureur fixe un prix pour le contrat d'assurance avant de savoir quel sera le coût de revient de cet engagement ; on parle d'inversion du cycle de production.

1.1.2 L'enjeu des provisions techniques en assurance IARD

L'inversion du cycle de production en assurance nécessite que l'assureur soit à même de gérer son risque. C'est la raison pour laquelle il est soumis à l'obligation de constituer des provisions techniques. Ces provisions techniques correspondent à l'ensemble des dettes

probables futures de l'assureur envers les bénéficiaires, au titre des engagements pris à la souscription du contrat, et doivent être inscrites au passif de son bilan.

La constitution des provisions est à la fois un gage de sécurité pour le client, et de stabilité du système financier. De plus l'évaluation et la constitution des provisions techniques est un enjeu stratégique pour la pérennité de l'activité de l'assureur. Le fait de ne pas sous-évaluer le montant des provisions lui permet d'éviter la ruine ou le manque de liquidité. Le fait de ne pas sur-évaluer le montant des provisions techniques lui permet une gestion d'entreprise plus rentable, où il n'immobilise pas des capitaux.

Compte tenu de l'importance de son poids dans le bilan, la problématique de provisionnement visant à l'estimer est donc un enjeu majeur pour l'assureur.

1.1.3 Les provisions techniques dans le cadre de notre étude

Le périmètre de notre étude est celui de l'assurance IARD. Cette abréviation, largement utilisée dans le monde de l'assurance, correspond aux termes **incendie**, **accidents** et **risques divers**. On dénombre plusieurs catégories de provisions techniques propres à l'activité d'assurance IARD. Dans le référentiel comptable français, nombre de ces provisions ne présentent aucune complexité en termes de gestion des risques car leur calcul est indiqué par des conventions de comptabilité, elles constituent toutefois des provisions nécessaires à un exercice prudent de l'activité d'assurance.

Dans le cadre de notre étude, nous nous intéressons exclusivement à la provision pour sinistre à payer (PSAP).

1.1.3.1 La provision pour sinistre à payer

La provision pour sinistres à payer n'est pas l'unique provision technique comptable qu'une compagnie d'assurance doit faire figurer à son bilan, en revanche elle constitue une part majeure des provisions techniques et est celle dont le calcul présente le principal enjeu. Cette provision correspond à l'ensemble des dépenses afférentes au règlement des sinistres par l'assureur au cours d'un exercice comptable. Elle peut être définie formellement comme le montant total en principal et en frais, tant internes qu'externes, nécessaire au règlement des sinistres prévu aux contrats, conformément à l'article R343-7 du Code des Assurances. Dans le cadre de notre étude nous ne nous intéresserons qu'au montant principal de la dette de l'assureur envers les bénéficiaires.

La visée du provisionnement étant d'estimer le coût de revient final du sinistre, il convient de s'intéresser au déroulement d'un sinistre et aux grandeurs que les assureurs emploient dans sa gestion.

1.1.3.2 Déroulement et gestion d'un sinistre

Nous introduisons dans la suite de cette section le vocabulaire relatif au traitement d'un sinistre IARD. Nous mettons en lien les notions que nous présentons avec le déroulement chronologique d'un sinistre.

La charge dossier/dossier

La charge d'un sinistre est un montant qui résulte d'une première estimation par un gestionnaire de sinistre. Cette estimation est basée sur des règles strictes, par exemple par une approche forfaitaire, un montant défini par les conventions, un coût moyen statistique ou encore en se référant à un barème de provisionnement. Elle est constituée à la déclaration du sinistre par une méthode d'évaluation dossier par dossier et vise à anticiper quel sera le montant total des paiements à la clôture du sinistre. Toute évolution de la charge du sinistre résulte d'une réévaluation à la hausse ou à la baisse par les gestionnaires de sinistre. Le montant de la charge peut être réévalué à tout instant notamment à l'aune de nouvelles informations.

Les paiements cumulés

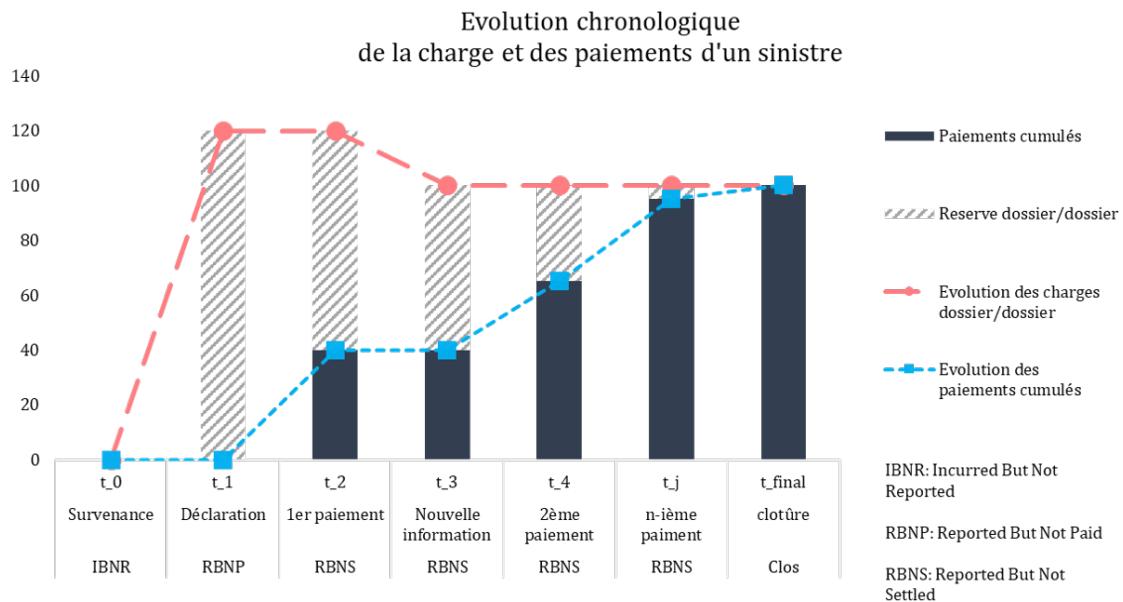
A une date t donnée, le montant des paiements cumulés représente le montant total des paiements versés au bénéficiaire de l'assurance au titre du sinistre. L'évolution du montant des paiements cumulés résulte d'un flux financier relatif au règlement du sinistre. Il peut s'agir d'un paiement réalisé par l'assureur ou encore d'un versement perçu par l'assureur, dans le cas d'un recours.

La réserve dossier/dossier

La réserve dossier/dossier correspond à la charge dossier/dossier résiduelle, c'est-à-dire la différence entre la charge dossier/dossier et les paiements cumulés.

On peut se fixer un horizon de temps dans le futur, que l'on nomme *ultime* et pour lequel ces grandeurs sont inconnues.

FIGURE 1.1 – Évolution chronologique de la charge et des paiements d'un sinistre.



Le provisionnement consiste donc en l'évaluation de la charge à l'ultime ou encore de la valeur des paiements cumulés à l'ultime. Le principal enjeu est de pouvoir prédire avec suffisamment d'exactitude les coûts futurs des sinistres et les montants à provisionner.

Cette tâche peut mobiliser des ressources différentes selon la norme à laquelle se plie l'assureur.

1.2 Les multiples approches de provisionnement selon les différentes normes

L'activité d'assurance est réglementée par un ensemble de dispositions, qui déterminent notamment les obligations de l'assureur en termes d'évaluation de ses provisions techniques. Selon la réglementation, l'évaluation de ces provisions techniques peut être réalisée par des approches différentes. Bien que les modalités d'estimation soient comparables, nous présentons les spécificités inhérentes à chaque cadre réglementaire.

1.2.1 Comptes sociaux : évaluation prudente

Lors de la réalisation du bilan comptable inclus dans les comptes sociaux de l'assureur, la provision pour sinistres à payer est déterminée par une évaluation dossier par dossier et en formulant une hypothèse prudente. Les articles 143-9 à 143-16 du règlement n° 2015-11 relatif aux comptes annuels des entreprises d'assurances de l'autorité des normes comptables , qui obéit au Code des Assurances, fixent le cadre de l'estimation et les éventuelles dispositions particulières.

L'évaluation repose généralement sur une méthode de provisionnement classique telle que la méthode de Chain-Ladder¹. Ces méthodes d'évaluation conduisent en général à l'obtention d'un montant moyen, mais peuvent aussi être employées pour obtenir une distribution empirique de plusieurs évolutions possibles du coût. Sur la base de ces estimations, on détermine un montant reflétant l'intégration d'une marge de prudence. Par exemple :

- Si l'on dispose de l'estimation de plusieurs valeurs possibles, on peut considérer un quantile empirique.
- Si l'on dispose d'un montant moyen, on peut ajouter un surplus sous la forme d'une marge forfaitaire (par exemple 20%), qui traduit la volonté de prudence.

Ainsi dans l'établissement des comptes sociaux, la constitution de provisions comptables est réalisée de façon prudente. Cette prudence peut mener à une sur-provision qui a principalement pour effet une anticipation du résultat erronée ainsi qu'un impact fiscal.

1. Nous présentons cette méthode de provisionnement dans une section ultérieure

1.2.2 Solvabilité 2 : évaluation économique *Best Estimate*

La norme Solvabilité 2 s'inscrit dans un cadre prudentiel où l'on s'intéresse aux risques auxquels est exposée l'entreprise et à la solvabilité de l'assureur. Entrée en application le 1er janvier 2016, elle propose d'établir un bilan en ayant une vision économique et non pas comptable.

Dans ce cadre, les provisions techniques représentent la valeur de transfert de l'entreprise, c'est-à-dire le montant qu'il faudrait donner à un tiers pour qu'il accepte de reprendre tous les engagements de l'entreprise. Les normes techniques d'exécution de Solvabilité 2, sont proposées par l'*EIOPA*².

L'évaluation des provisions techniques IARD consiste en un calcul de la meilleure estimation, le *Best Estimate (BE)*, calculé comme la somme des *cash-flows* futurs actualisés, auquel on ajoute une marge pour le risque, la *Risk Margin (RM)*.

$$\text{provisions techniques} = \frac{\text{flux de trésorerie futurs actualisés}}{BE} + \frac{\text{marge pour le risque}}{RM}$$

La provision comptable *PSAP* trouve son correspondant dans la norme Solvabilité 2 en la valeur du *BE* de réserves.

Une estimation de la valeur des flux de trésorerie futurs est calculée en utilisant une méthode de provisionnement déterministe ou stochastique. On peut notamment recourir à une méthode de provisionnement classique de type Chain-Ladder. Pour l'actualisation des flux, on utilise la courbe réglementaire des taux sans risque fournie par l'*EIOPA*.

La marge pour le risque *RM* est quant à elle calculée par l'approche coût du capital, en projetant le capital réglementaire, appelé *Solvency Capital Requirement (SCR)*. Le *SCR* est déterminé en considérant la variation des fonds propres à 1 an et calculant la *Value at Risk* de la variable de perte au seuil de 99,5%. Les entreprises ont deux possibilités pour le calcul du capital réglementaire : elle peuvent appliquer la formule standard proposée par le régulateur ou bien développer leur propre modèle interne.

Ainsi, dans l'approche Solvabilité 2, bien qu'il faille compléter le *BE* par le calcul d'un *SCR* et d'une marge pour le risque *RM*, il est essentiel de disposer d'un modèle de provisionnement permettant d'estimer avec justesse les flux de trésorerie futurs pour l'évaluation des provisions techniques.

2. *EIOPA* : Autorité européenne des assurances et des pensions professionnelles

1.2.3 IFRS17 : vision consolidée des comptes

La norme IFRS17 s'inscrit dans un cadre financier où l'on s'intéresse à la comparabilité des entreprises et à la transparence des comptes. Publiée en mai 2017 par l'IASB³, la norme entrera en vigueur le 1^{er} Janvier 2022.

Dans ce cadre, en ce qui concerne le périmètre IARD, pour des polices d'une durée de couverture inférieure à 12 mois, les provisions techniques se présentent comme la somme d'une estimation des flux de trésorerie futurs et d'un ajustement pour le risque, *RA*.

$$\text{provisions techniques IARD} = \frac{\text{flux de trésorerie futurs}}{\text{estimation}} + \frac{\text{ajustement pour le risque}}{RA}$$

L'évaluation des passifs des contrats d'assurance peut être réalisée selon trois modèles. Pour ce qui est du périmètre IARD, on emploie en général soit l'approche dite générale⁴ soit une approche simplifiée dite d'allocation de primes⁵. Ces deux approches nécessitent toutes deux de calculer une juste estimation des flux de trésorerie futurs par une méthode de provisionnement.

Ainsi, bien qu'il faille compléter l'estimation des flux de trésorerie futurs par le calcul d'un ajustement pour le risque, la détention d'une méthode de provisionnement qui nous permette d'estimer les flux futurs est également nécessaire dans le cadre IFRS17.

Notre présentation des différentes exigences d'un assureur en matière de provisionnement nous a permis de mettre en évidence que l'estimation des flux de trésorerie futurs reste une étape cruciale dans l'évaluation des provisions, bien qu'il faille selon le cadre procéder à des ajustements (intégration de marge, actualisation, ... etc).

Ce constat souligne l'intérêt que revêt la détention d'une méthode de provisionnement pour estimer les flux futurs relatifs au sinistres assurés, comme nous proposons de nous y intéresser dans notre étude.

Plusieurs modèles de provisionnement existent et peuvent être employés pour réaliser la meilleure estimation des flux futurs. Parmi ceux-ci, le modèle de Mack-Chain-Ladder est le plus largement utilisé.

3. IASB : *International Accounting Standards Board*, bureau international des normes comptables

4. *BBA* : Building Block Approach

5. *PPA* : Premium Allocation Approach

Chapitre 2

La réalisation du provisionnement par les actuaires

2.1 Un modèle classique de provisionnement : Mack-Chain-Ladder

La méthode de Chain-Ladder est une méthode de provisionnement de référence largement utilisée par les assureurs, aussi bien pour estimer un montant de paiements cumulés qu'un montant de charges. En 1993 Thomas Mack se propose de formaliser mathématiquement la méthode de Chain-Ladder et présente un modèle stochastique qui en offre une généralisation (MACK, 1993).

Nous présentons ce modèle dans cette section. Nous formaliserons dans un premier temps la représentation des données sur laquelle la méthode de Chain-Ladder s'appuie, nous présenterons ensuite le modèle et sa mise en œuvre, enfin nous discuterons des avantages et des limites du modèle. Bien que nous pourrons être amenée à employer le terme Chain-Ladder par un abus de langage, c'est le modèle de Mack-Chain-Ladder que nous considérerons tout au long de notre étude.

2.1.1 Formalisation préliminaire : représentation d'un triangle de sinistres

La méthode de développement de Chain-Ladder repose sur une représentation de la chronologie du règlement des sinistres sous la forme d'un tableau à double entrée.

Ce tableau représente des sinistres à un niveau d'agrégation¹ choisi par la compagnie. Les informations contenues dans le tableau peuvent être des montants de paiements cumulés, des montants de charges ou encore un nombre de sinistres. Chaque agrégat de sinistres est représenté par une ligne du tableau, qui nous permet de lire pour chacun :

- l'année de rattachement i des sinistres, pour $i \in [0, I]$ dans la première colonne
- les montants $C_{i,j}$ des paiements cumulés ou de charges après un délai de développement de j années pour $j \in [0, J]$ dans chacune des colonnes suivantes

1. Le choix du niveau d'agrégation des sinistres pour la tâche de provisionnement n'est pas l'enjeu du mémoire.

Le choix de l'année de rattachement du sinistre est important. En assurance IARD, l'année de rattachement du sinistre est très souvent l'année de survenance, aussi notre périmètre étant celui-ci, nous pourrons nous référer à l'année de rattachement par la dénomination année de survenance. On peut toutefois noter qu'il existe certaines branches pour lesquelles l'année de rattachement peut être l'année de déclaration², ou encore l'année de souscription³. Les années de développement s'expriment en ajoutant un délai de j années à partir de l'année de rattachement.

A une date donnée, les montants futurs n'étant pas connus, le tableau se présente comme l'association d'un triangle supérieur contenant les informations historiques relatives à la sinistralité, et d'un triangle inférieur initialement vide, que l'on cherche à compléter. Ce triangle nous montre l'évolution de la liquidation des sinistres. On parle ainsi de triangle de liquidation.

TABLE 2.1 – Exemple de triangle de liquidation.

I=4, J=4

$i \backslash j$	0	1	2	3	4
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	$C_{0,3}$	$C_{0,4}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	
2	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$		
3	$C_{3,0}$	$C_{3,1}$			
4	$C_{4,0}$				

La valeur $C_{0,3}$ représente le montant total en date de l'année (0+3) pour les sinistres survenus au cours de l'année 0.

La valeur $C_{0,4}$ représente l'évolution, du montant $C_{0,3}$ une année plus tard

Le provisionnement a pour objectif d'estimer les montants $C_{i,J} \forall i \in [0, I]$. La valeur J est un horizon de temps au delà duquel on suppose que les sinistres sont clos. On s'y réfère par le terme *ultime*. Les montants $C_{i,J} \forall i \in [0, I]$ sont alors désignés comme étant les montants à l'ultime.

Le choix de la valeur J dépend très fortement du type d'activité assurantielle. Pour une branche d'activité IARD donnée, l'horizon de temps J entre la survenance et la clôture du sinistre permet de qualifier la branche, selon la longueur du développement. Ainsi une branche courte, est une branche dont la cadence de développement est rapide avec une durée de développement J que l'on peut considérer comme étant petite. Par exemple, l'assurance dommage automobile est généralement considérée comme une branche courte par opposition à l'assurance responsabilité civile automobile qui est longue⁴. De ce fait,

2. Certaines garanties RC comme la RC médicale par exemple

3. En assurance caution par exemple

4. Notamment parce qu'il y a des enjeux juridiques de responsabilité et des difficultés à évaluer précisément le préjudice, ce qui conduit à un délai plus long pour la résolution.

ces branches nécessitent de se projeter sur des horizons de temps qui sont de différents ordres de grandeurs.

Pour ce qui est de la valeur de I , il est pertinent de remarquer que si l'on dispose d'un historique suffisant, on peut toujours se ramener au cas où $I = J$; on peut ainsi noter N l'entier tel que $I = J = N$.

Ayant ainsi formalisé la représentation de sinistres sous forme de triangles nous présentons le modèle de Mack-Chain-Ladder, dont la mise en œuvre repose sur cette visualisation.

2.1.2 Mise en œuvre du modèle de Mack-Chain-Ladder

Ayant introduit la représentation des sinistres sous forme de triangles de liquidation, nous proposons dans cette section une mise en œuvre du modèle de provisionnement de Mack-Chain-Ladder, qui emploie cette description. Le contenu des triangles de liquidation sera supposé être le montant des paiements cumulés. De plus on supposera que les entiers I et J ont même valeur, on notera N l'entier tel que $I = J = N$.

2.1.2.1 Description du modèle

Dans le cadre de son modèle, Mack considère les montants $C_{i,j}$ comme des réalisations de variables aléatoires. Il formule trois hypothèses devant être vérifiées par ces montants.

1. Indépendance des montants cumulés des sinistres pour des années de survenance distinctes.

$$\forall i, l \in \{0, \dots, N\}, i \neq l \Rightarrow \quad \forall j \in \{0, \dots, N\}, C_{i,j} \perp\!\!\!\perp C_{l,j} \quad (2.1)$$

2. Existence de facteurs de développements f_j qui permettent d'exprimer l'espérance d'un montant $C_{i,j}$ conditionnellement aux observations antérieures. De plus, chacun de ces facteurs de développement ne dépend que de l'année de développement j considérée.

$$\forall j \in \{1, \dots, N-i\}, \exists f_j / \quad \mathbb{E}[C_{i,j+1}|C_{i,1}, \dots, C_{i,j}] = f_j C_{i,j} \quad (2.2)$$

3. Expression de la variance des $C_{i,j}$ conditionnellement aux observations antérieures.

$$\forall j \in \{1, \dots, N\}, \exists \sigma_j > 0 / \quad \mathbb{V}[C_{i,j+1}|C_{i,1}, \dots, C_{i,j}] = \mathbb{V}[C_{i,j+1}|C_{i,j}] = \sigma_j^2 C_{i,j} \quad (2.3)$$

Sous ces hypothèses, on peut déterminer l'espérance de la charge à l'ultime conditionnellement aux données historiques contenues dans le triangle supérieur $\mathcal{F}_N = \{C_{i,j}/i+j \leq N\}$

$$\mathbb{E}[C_{i,N}|\mathcal{F}_N] = \left(\prod_{k=N-i}^{N-1} f_k \right) C_{i,N-i}$$

Les données de \mathcal{F}_N nous permettent de construire un estimateur non biaisé \hat{f}_j des facteurs de développement f_j qui permettent de passer d'une année à la suivante.

En effet pour une année de développement $j \in \{0, \dots, N-1\}$,

$$\forall i \in \{0, \dots, N\}, f_j = \frac{\mathbb{E}[C_{i,j+1} | C_{i,1}, \dots, C_{i,j}]}{C_{i,j}}.$$

On détermine donc un estimateur de f_j à partir des données de paiements antérieurs.

$$\hat{f}_j = \frac{\sum_{i=1}^{N-j} C_{i,j+1}}{\sum_{i=1}^{N-j} C_{i,j}}$$

Cet estimateur des facteurs de développement est utilisé pour calculer une estimation des paiements cumulés à l'ultime :

$$\hat{C}_{i,N} = \left(\prod_{k=N-i}^{N-1} \hat{f}_k \right) C_{i,N-i}$$

Mack propose également un estimateur sans biais des coefficients de variance $(\sigma_j)_{j \in \{1, \dots, N\}}$

$$\hat{\sigma}_j^2 = \frac{1}{N-j-1} \sum_{i=0}^{N-j-1} C_{i,j} \left(\frac{C_{i,j+1}}{C_{i,j}} - \hat{f}_j \right)^2$$

2.1.2.2 Application sur un exemple

TABLE 2.2 – Application de la méthode Chain-Ladder à un exemple.

		Triangle à compléter paiements cumulés (exemple)					Estimateurs des facteurs de développement	
		0	1	2	3	4	coefficient	valeur*
		i	j				\hat{f}_j	
	0	55	97	134	162	171	\hat{f}_0	1,77
	1	60	107	143	170		\hat{f}_1	1,37
	2	66	118	163			\hat{f}_2	1,20
	3	62	109				\hat{f}_3	1,06
	4	65					*valeurs arrondies.	

Les informations historiques contenues dans le triangle à compléter nous permettent de procéder au calcul des coefficients de développement pour chaque année $j \in [0, 3]$. Munis de la valeur de ces coefficients on peut procéder à une estimation des paiements cumulés à $j = 4$ pour toutes les années de survenance.

TABLE 2.3 – Prévisions obtenues à l’ultime par la méthode Chain-Ladder.

$i \backslash j$	0	1	2	3	4
0	55	97	134	162	171
1	60	107	143	170	179
2	66	118	163	195	206
3	62	109	149	179	188
4	65	115	158	189	199

2.1.3 Limites et remise en cause de Chain-Ladder

La méthode de Chain-Ladder que nous avons présentée est aujourd’hui considérée comme une méthode de référence pour le provisionnement IARD, à l’aune de laquelle on peut jauger les performances d’autres méthodes.

Elle possède l’avantage d’une très grande simplicité de mise en œuvre des calculs, ainsi que d’une polyvalence puisqu’elle peut être utilisée sur différents type de triangles : triangles de charges, triangles de règlements, triangles de nombres de sinistre, ou encore un triangle où sont inscrits les coûts moyens de sinistres, obtenus en divisant le montant $C_{i,j}$ par le nombre de sinistres auxquels il se rapporte.

Toutefois, la validité et la pertinence de ses résultats dépendent d’hypothèses fortes. Notamment, l’hypothèse de constance du ratio $\frac{C_{i,j+1}}{C_{i,j}}$ est très souvent loin d’être satisfaite. Un autre problème que pose l’utilisation de la méthode de Chain-Ladder est la très forte sensibilité des coefficients à la volatilité de la donnée. Ce fait induit une volatilité des coefficients, qui rajoute beaucoup d’incertitude sur l’estimateur. De plus, la méthode de Chain-Ladder est très sensible aux observations atypiques et aux valeurs aberrantes qui nécessitent l’intervention d’un actuaire afin de réaliser des retraitements. Un quatrième point est une incohérence des coefficients en cas de changement dans l’activité de l’assureur tel que l’évolution de son barème d’indemnisation, qui introduirait un biais. En outre on peut également noter qu’il est difficile d’appliquer la méthode de Chain-Ladder pour des lignes de business où l’on est confronté à une insuffisance de l’historique. Il faut alors songer à des façons de prolonger le développement artificiellement. Enfin, le principe de la méthode ignore la typologie des business auxquels se rattachent les triangles sur lesquels on l’applique, et ne tient pas compte de l’historique de développement du sinistre puisqu’il se base uniquement sur les montants à une seule date. Cette limite force à l’utilisation de méthodes alternatives basée sur des *a priori* telle que la méthode de Bornhuetter-Ferguson (BORNHUECKER & FERGUSON, 1972).

Face à ces limites de nouveaux modèles de provisionnement sont proposés. Plusieurs d’entre eux suggèrent l’emploi de méthodes s’appuyant sur le *Deep Learning*.

2.2 L'essor de méthodes de provisionnement reposant sur le *Deep Learning*

Dans le domaine de l'assurance en général, les réseaux neuronaux ont été proposés afin de traiter des problématiques d'assurance santé, de détection de fraudes, notamment en assurance automobile, ou encore de prédiction d'insolvabilité des assureurs (BROCKETT et al., 1997). Les réseaux de neurones ont également été suggérés dans des problématiques de provisionnement. Nous avons mené une revue de littérature qui nous a permis de le documenter et d'établir un état des lieux des publications sur le sujet.

2.2.1 Un intérêt croissant

Les méthodes de provisionnement s'appuyant sur le *Deep Learning* ne se sont pas apparues récemment. Ainsi il y a 15 ans, BRAUN et LAI (2005) ou encore MULQUINEY (2006) proposaient déjà des méthodes de provisionnement s'appuyant sur l'utilisation des réseaux de neurones artificiels.

Toutefois, on peut observer l'intérêt particulier qu'elles suscitent actuellement au sein de la profession actuarielle. On constate notamment leur présence dans des publications de référence à visée internationale telles que les bulletins bi-annuels de la section ASTIN⁵ de l'association internationale des actuaires (*IAA*). Dans la publication de 2018 (JAMAL et al., 2018), les méthodes s'appuyant sur les réseaux de neurones figurent ainsi au côté des méthodes de provisionnement classiques dans le panorama dressé par l'ASTIN.

L'intérêt croissant pour ces modèles est associé à un nombre plus important de publications au cours des trois dernières années. Ces publications marquent un renforcement de la volonté de recourir à des méthodes dont l'utilisation est encore peu répandue dans le domaine du provisionnement. On peut y voir une volonté de tirer parti des avancées en *machine learning* pour proposer de nouveaux modèles de provisionnement.

2.2.2 Des publications marquées au cours des trois dernières années

En 2017 l'ASTIN présente notamment dans une publication dédiée, une architecture de réseau de neurones, fruit de son travail de recherche, ainsi qu'un algorithme d'optimisation à employer pour calibrer le modèle (HAREJ et al., 2017). Le modèle présenté propose un développement des triangles de liquidation par un mécanisme de prédiction en cascade permettant de reproduire les étapes de la méthode de Chain-Ladder. Un réseau de neurones distinct est entraîné pour chaque année de développement.

5. ASTIN : Actuarial Studies in Non Life

En 2018 Mario Wüthrich propose une méthode de provisionnement mettant en œuvre les réseaux de neurones à des fins de provisionnement à une maille d'agrégation fine, en assurance accident (WÜTHRICH, 2018). Son modèle, s'appuie sur l'utilisation de données simulées (GABRIELLI & WÜTHRICH, 2018).

Parmi les publications récentes, on peut également noter GABRIELLI, 2019, qui propose d'employer un réseau de neurones afin de *booster* un modèle de Poisson sur-dispersé d'estimation de la fréquence des sinistres.

Enfin en 2018 Kevin Kuo propose un modèle de provisionnement s'appuyant sur un type de réseau de neurones particulier, *Deep Triangle* (KUO, 2018). Ce modèle se distingue notamment des autres par l'emploi de structures récurrentes. Kevin Kuo en illustre les performances par une analyse comparative avec d'autres méthodes stochastiques et déterministes.

Ainsi, plusieurs modèles de provisionnement publiés récemment mettent en œuvre le *Deep Learning* et ont motivé notre étude. Avant de procéder à une étude approfondie de l'un de ces modèles (KUO, 2018), nous proposons une présentation des outils de *Deep Learning* qu'ils mettent en œuvre.

Chapitre 3

Les réseaux de neurones

Dans ce chapitre nous proposons une présentation progressive du fonctionnement des outils que nous avons employés dans notre étude, que nous amorçons dans un premier temps avec la présentation du neurone unique, structure élémentaire à partir de laquelle tout réseau de neurones est constitué. Le modèle du neurone unique nous permettra dans un deuxième temps de présenter la mise en œuvre de l'apprentissage. Nous illustrerons ensuite la nécessité de composer des réseaux associant plusieurs neurones et montrerons les capacités d'approximation de fonction des réseaux de neurones. Finalement, nous aboutirons à la présentation d'une catégorie de réseaux de neurones spécifiques dont nous avons fait usage : les réseaux de neurones récurrents.

3.1 Présentation du neurone unique : formalisation mathématique

Un neurone se présente comme une application mathématique qui permet d'observer une distinction selon un critère vérifié ou non par l'argument qu'on lui fournit. L'argument de l'application est nommée entrée du neurone, et le résultat de l'application est la sortie. Un neurone reçoit en entrée un ensemble d'informations et produit en sortie une valeur unique.

L'application mathématique que constitue un neurone est caractérisée par :

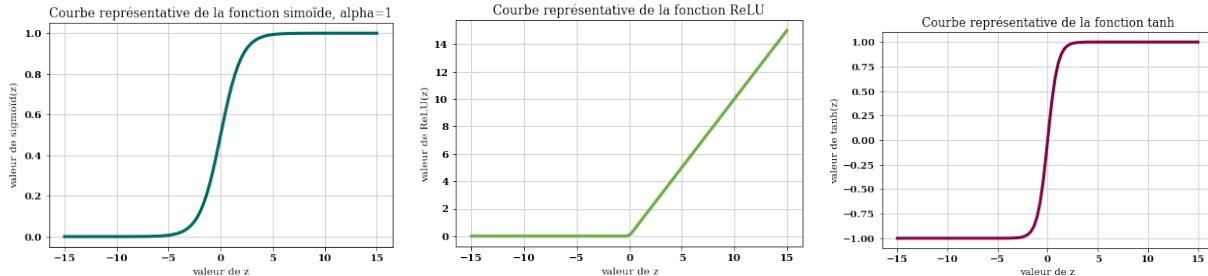
- **Un espace de départ** \mathcal{X} auquel appartient l'information d'entrée X . En général il existe $n \in \mathbb{N}^*$ tel que \mathcal{X} est isomorphe à un sous-espace de \mathbb{R}^n . On identifie l'information X et son image par cet isomorphisme $(X_i)_{1 \leq i \leq n} \in \mathbb{R}^n$.
Pour chaque élément $X \in \mathcal{X}$ nous introduisons la notation \bar{X} pour désigner le vecteur de \mathbb{R}^{n+1} tel que $\bar{X}_0 = -1$ et $\forall i \in [1, n] \bar{X}_i = X_i$.
- **Un espace d'arrivée** \mathcal{Y} auquel appartient l'information de sortie y , en général \mathcal{Y} est une partie de \mathbb{R} . Lorsque \mathcal{Y} est un ensemble fini, alors la tâche réalisée par le neurone est une tâche de classification. Le cas où \mathcal{Y} n'est pas fini correspond à une tâche de régression.

- **Un ensemble de poids** $W = (w_i)_{0 \leq i \leq n} \in \mathbb{R}^{n+1}$, grandeur à calibrer dont la valeur détermine entièrement les prédictions du neurone. On note w le vecteur $(w_i)_{1 \leq i \leq n}$, c'est à dire W privé de w_0 .
- **Une fonction d'activation** $\phi : \mathbb{R} \rightarrow \mathbb{R}$, qui intervient dans la réponse du neurone. En général la fonction d'activation ϕ vérifie les hypothèses suivantes :
 1. ϕ est monotone croissante
 2. ϕ est différentiable presque-partout
 3. ϕ est à valeurs dans un intervalle fermé ou dans un ensemble fini de \mathbb{R} .

TABLE 3.1 – Présentation de trois exemples de fonctions d'activation.

Fonction	sigmoïde, $\alpha \in \mathbb{R}_+$	ReLU	tanh
Domaine de valeurs	$[0,1]$	\mathbb{R}_+	$[-1,1]$
Expression	$z \mapsto \frac{1}{1+e^{-\alpha z}}$	$z \mapsto \max(0, z)$	$z \mapsto \frac{e^z - e^{-z}}{e^z + e^{-z}}$

FIGURE 3.1 – Courbes représentatives de trois fonctions d'activation pouvant être utilisées.



Résultat fourni par le neurone

Soit $X = (X_i)_{1 \leq i \leq n}$, un vecteur représentant l'ensemble des informations fournies en entrée du neurone.

Les poids du neurone sont tels que pour tout i le poids w_i est lié à l'information X_i . Le poids w_0 est le coefficient de biais, également appelé seuil. On l'associe à une information X_0 constante égale à -1.

Le neurone simple opère une somme pondérée des informations et retourne la sortie :

$$y = \phi \left(\sum_{i=0}^n w_i * X_i \right)$$

Ainsi pour chaque ensemble d'information X fourni en entrée du neurone, le neurone retourne le résultat

$$\begin{aligned} y &= \phi(w^t X - w_0) \\ y &= \phi(W^t \bar{X}) \end{aligned}$$

On note

$$y = F_W(X)$$

3.2 Entraînement du neurone unique pour une tâche de prédiction

L'entraînement du neurone est l'étape au cours de laquelle on détermine la valeur optimale des poids W^{opt} pour une tâche. Il est réalisé par apprentissage statistique supervisé qui correspond à la situation où l'on utilise un sous-échantillon pour lequel on a déjà les bonnes réponses, afin de pouvoir construire un outil qui nous permet de prédire les réponses pour de nouvelles données. Pour le mettre en œuvre, on doit disposer d'un échantillon de données labellisées, c'est-à-dire de données associées à des étiquettes de valeurs leur correspondant.

La réalisation de l'entraînement consiste en la résolution d'un programme d'optimisation par laquelle on détermine les poids W^{opt} qui minimisent l'erreur de prédiction du neurone.

La mise en œuvre de l'entraînement implique donc, d'une part, de pouvoir mesurer l'erreur de prédiction du modèle. Pour cela il nous faut en premier lieu disposer d'informations sur les vraies valeurs, et en second lieu définir une mesure d'erreur à retenir. D'autre part, une fois l'erreur calculée, il nous faut choisir un algorithme d'optimisation pour la minimiser.

3.2.1 L'échantillon d'entraînement

L'entraînement d'un neurone par apprentissage supervisé nécessite de disposer d'un échantillon de données étiquetées :

$$\mathcal{D} = \{(X^k, y_k), k \in [1, K]\}$$

\mathcal{D} est un ensemble de K couples $(X^k, y_k) \in \mathcal{X} \times \mathcal{Y}$, où pour tout k , l'étiquette y_k est la vraie valeur correspondant à X^k .

Pour chacun des couples (X^k, y_k) , on note \hat{y}_k la valeur prédictée par le neurone. On peut faire le lien avec les notations introduites précédemment et préciser que $\hat{y}_k = F_W(X^k)$.

3.2.2 La fonction de perte et la fonction de coût

La fonction de perte l est une fonction de deux variables réelles qui permet de calculer une erreur de prédiction du neurone sur l'échantillon d'entraînement \mathcal{D} . Pour une fonction de perte l , l'erreur de prédiction du neurone sur l'échantillon d'entraînement, aussi appelée fonction de coût, s'exprime ainsi :

$$E(\mathcal{D}, W) = \frac{1}{K} \sum_{k=1}^K l(y_k, \hat{y}_k) \quad \text{où } l \text{ est la fonction de perte}$$

Il est fondamental de noter que la fonction de coût est une fonction du vecteur de poids W , puisque $\hat{y}_k = F_W(X^k) = \phi(W^t \overline{X^k})$. On adopte la notation $E^{\mathcal{E}}$ pour la fonction de coût calculée sur les éléments d'un ensemble \mathcal{E} .

Plusieurs choix sont possibles pour la fonction l , le choix d'une fonction de perte dépendant principalement du type de problème que l'on cherche à résoudre. Dans le cas de la tâche de provisionnement à laquelle nous avons consacrée notre étude, $\mathcal{Y} = \mathbb{R}$ ce qui correspond à une tâche de régression, les exemples suivants sont alors des choix appropriés.

TABLE 3.2 – Exemples de fonctions de perte pour la régression et expression de la fonction de coût associée.

Expression de la fonction de perte l : $(y_k, \hat{y}_k) \mapsto l(y_k, \hat{y}_k)$			
$(y_k - \hat{y}_k)^2$	$ y_k - \hat{y}_k $	$y_k \cdot \ln(\hat{y}_k) + (1 - y_k) \cdot \ln(1 - \hat{y}_k)$	
Expression de la fonction de coût	$\frac{1}{K} \sum_{k=1}^K (y_k - \hat{y}_k)^2$	$\frac{1}{K} \sum_{k=1}^K y_k - \hat{y}_k $	$\frac{1}{K} \sum_{k=1}^K y_k \cdot \ln(\hat{y}_k) + (1 - y_k) \cdot \ln(1 - \hat{y}_k)$
Type d'erreur	erreur moyenne quadratique (mse)	erreur moyenne absolue (mae)	entropie croisée

Le choix d'une fonction de perte nous permet d'exprimer explicitement la fonction de coût à minimiser. En pratique lorsque l'on vient à implémenter un réseau de neurones et à choisir ses paramètres d'entraînement, on renseigne directement la fonction de coût.

3.2.3 L'algorithme d'optimisation : la descente de gradient

L'entraînement du neurone se ramène à la résolution du programme d'optimisation suivant :

$$\min_W E^{\mathcal{D}}(W) = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} l(y_k, \phi(W^t \bar{X}^k))$$

où $|\mathcal{D}|=K$, est le cardinal de l'échantillon \mathcal{D} .

Pour le réaliser, on emploie un algorithme d'optimisation appartenant à la famille des descentes de gradient qui conduit à l'obtention de poids optimaux W^{opt} .

Optimisation par descente de gradient stochastique

On a recours à un algorithme de descente de gradient stochastique pour résoudre le problème d'optimisation. La méthode consiste en une minimisation itérative, qui permet de converger vers un minimum qui peut n'être que local.

Nombre d'époques

Afin d'obtenir de bons résultats lorsque l'on réalise l'optimisation par descente de gradient stochastique, il est parfois nécessaire d'utiliser les données d'entraînement plusieurs fois : on parle de réalisation de plusieurs époques. Une époque correspond à l'utilisation de chacune des données de \mathcal{D} une seule fois. La réalisation de plusieurs époques permet d'assurer la convergence vers un minimum local. Le nombre d'époques à réaliser est un hyperparamètre de l'entraînement à choisir, et dont la valeur est susceptible d'influer sur les capacités de prédiction du neurone.

Nombre de *batchs* par époques

Au début de chaque époque, l'échantillon d'entraînement \mathcal{D} est divisé aléatoirement en plusieurs lots, appelés *batchs*. Chaque *batch* sera utilisé au cours d'une itération. La taille d'un *batch*, et de façon équivalente, le nombre de *batchs* par époque, peuvent être choisis au moment de l'entraînement. L'utilisation d'un sous-échantillon à chaque itération permet d'adapter l'algorithme de descente de gradient général à des volumes importants de données.

Nombre d'itération total au cours de l'entraînement

Le nombre de *batchs* par époque et le nombre d'époques à réaliser permettent de calculer le nombre d'itération T dans le processus d'optimisation par descente de gradient.

Description du processus d'optimisation des poids

A chaque itération, on sélectionne un *batch*. On considère alors l'erreur de prédiction du neurone sur le *batch*, et on calcule le gradient de la fonction de coût par rétro-propagation (sous-section 3.3.2). Compte tenu de la valeur du gradient, les poids du neurone sont corrigés et mis à jour de façon à évoluer vers l'opposé du gradient. On sélectionne ensuite un nouveau *batch* pour une nouvelle itération. On réitère la correction des poids jusqu'à la

T^{ième} itération.

Une description formelle de la descente de gradient stochastique :

- 0 Initialisation arbitraire W_0
à l'instant t : on sélectionne un batch \mathbf{b}
- 1 On considère l'erreur $E^{\mathbf{b}}(W_t) = E_t(W_t)$

- 2 On calcul du gradient $\nabla E_t = \left(\frac{\partial E_t}{\partial w_0}, \frac{\partial E_t}{\partial w_1}, \dots, \frac{\partial E_t}{\partial w_K} \right)^T$

Les composantes du gradient s'expriment en fonction de la valeur à l'instant t des poids W_t , de la dérivée de la fonction d'activation, ainsi que des éléments de l'échantillon \mathcal{D} appartenant au batch \mathbf{b} .

- 3 Mise à jour des poids $W_{t+1} = W_t - \eta_t * \nabla E_t$
incrémentation : instant=t+1

Le coefficient η_t est le taux d'apprentissage, *learning rate*, qui est un paramètre important dans la descente de gradient : si la valeur du *learning rate* est faible, alors la convergence vers un extremum peut être très lente et nécessiter un nombre d'itérations très important. Une valeur trop importante en revanche, peut nous conduire à dépasser un minimum global ou même à diverger (BENGIO, 2012).

Réalisée avec les hyperparamètres appropriés, l'optimisation par descente de gradient nous permet de converger vers une valeur des poids minimisant l'erreur, et donc optimale, au sens local, dans la résolution de notre problème. A l'issue de la phase d'entraînement, le neurone est entraîné et on peut l'employer afin de réaliser des prédictions à partir de nouvelles données \mathcal{X} .

Nous avons ainsi mis en lumière le fonctionnement du neurone unique. Toutefois, les neurones uniques sont en général utilisés dans des architectures plus complexes : les réseaux de neurones. Aussi il convient de nous intéresser aux raisons pour lesquelles on emploie ces structures et à leurs capacités.

3.3 L'utilisation des réseaux de neurones

3.3.1 Les limites du neurone unique

Le neurone unique n'est pas sans intérêt ; utilisé seul, il permet notamment d'exécuter l'algorithme du perceptron simple introduit en 1958 (ROSENBLATT, 1958). Cet algorithme apprend une séparation de l'espace des données d'entrée selon un hyperplan et permet la classification binaire. Toutefois, le neurone unique présente la principale limite de ne pouvoir approximer qu'une famille restreinte de fonctions, comme l'on mis en évidence MINSKY et PAPERT (1969). L'un des écueils notable étant la fonction *OU exclusif, XOR*.

Des travaux de recherche ont permis de mettre en évidence les capacités d'une association de plusieurs neurones à résoudre les problèmes qui n'étaient pas solubles avec un neurone unique¹. Ces résultats ont mis en évidence que l'usage de structures moins élémentaires associant de multiples neurones permet d'élargir l'ensemble des fonctions qu'il est possible d'approximer, justifiant ainsi que leur usage soit privilégié au neurone simple. Ces structures, auxquelles on a recours qui associent plusieurs neurones uniques communiquant entre eux, sont des réseaux de neurones.

3.3.2 Présentation des réseaux de neurones

Les réseaux de neurones se présentent comme une association de neurones uniques au sein de laquelle l'information obtenue en sortie d'un neurone est employée comme information d'entrée d'un autre neurone.

Un réseau est composé de plusieurs couches de neurones : une couche d'entrée, une ou plusieurs couches intermédiaires dites "couches cachées" et une couche de sortie. Les neurones appartenant à une même couche possèdent la même fonction d'activation et ne communiquent pas entre eux. Ils échangent des informations avec les neurones de couches suivantes et les neurones des couches précédentes. Le sens de propagation de l'information dans un réseau n'est pas contraint.

Pour un réseau de neurones donné comportant un nombre L de couches, on peut associer à chaque couche l une fonction de transfert $\Phi^{[l]}$ ainsi qu'une matrice de poids $W^{[l]}$ de sorte que les lignes de $W^{[l]}$ correspondent aux vecteurs des poids des neurones de la couche l . La matrice de poids associée à la couche l est de dimension $q \times (p + 1)$ où le nombre de lignes q de la matrice est égal au nombre de neurones de la couche l , et p est la dimension des informations fournies en entrée des neurones de la couche. Une couche l du réseau de neurones reçoit en entrée une information I^l et retourne en sortie $F_{W^{[l]}}(I^l)$.

Ainsi, défini formellement, un réseau multicouches est caractérisé par :

- Un espace de départ \mathcal{X} auquel appartient l'information d'entrée, en général $\exists n \in \mathbb{N}^*$ tel que \mathcal{X} est isomorphe à un sous-espace de \mathbb{R}^n .
- Un espace d'arrivée $\mathcal{Y} \subset \mathbb{R}^m$, où $m \in \mathbb{N}^*$, auquel appartient l'information de sortie.
- Un nombre de couches L .
- Un vecteur indiquant la taille de chaque couche : $(t_l)_{1 \leq l \leq L}$.
- Un vecteur de matrices $W = (W^{[l]})_{1 \leq l \leq L}$ où $\forall l$, $W^{[l]}$ est la matrice associée la couche l . Le vecteur W correspond à l'ensemble des poids du réseau de neurones, sa valeur détermine entièrement les prédictions.
- Un ensemble de fonctions d'activation $(\Phi^{[l]})_{1 \leq l \leq L}$.

L'emploi d'un réseau de neurones a pour objectif d'approximer une fonction $f(\cdot)$ par une fonction paramétrique $\hat{f}(\cdot, W)$ où W est un paramètre à optimiser par apprentissage

1. Dans le cas de la fonction *XOR* une estimation est possible par une association de 3 neurones.

statistique sur un échantillon d’entraînement. On peut exprimer $\hat{f}(., W)$ à l’aide les fonctions $(F_{W^{[l]}})_{1 \leq l \leq L}$:

$$\hat{f}(., W) = F_{W^{[L]}} \circ F_{W^{[L-1]}} \circ \dots \circ F_{W^{[1]}}$$

L’optimisation est effectuée par descente de gradient. Le gradient est calculé par rétro-propagation (RUMELHART et al., 1986). Pour ce faire, on procède successivement aux calculs de gradient par application du théorème de dérivation des fonctions composées. Ainsi, la mise en œuvre de la rétro-propagation à l’échelle d’un réseau est menée en commençant par mettre à jour les poids de la dernière couche et en remontant jusqu’à la première couche.

3.3.3 Capacité d’estimation des réseaux de neurones

Contrairement au neurone simple dont nous avons exhibé les limites, les réseaux de neurones multicouches montrent des facultés de résolution de problème beaucoup plus étendues que des structures élémentaires. Les travaux de HORNICK et al. (1989) ont démontré le résultat suivant :

Théorème d’approximation universelle. Toute fonction mesurable définie sur un espace de dimension finie isomorphe à \mathbb{R}^n et à valeurs dans un espace de dimension finie isomorphe à \mathbb{R}^m peut être approximée d’autant près que l’on souhaite, au sens de la norme \mathbb{L}^p , par un réseau de neurones *feedforward* possédant une seule couche cachée, sans restrictions particulières sur la fonction d’activation.

L’espace des réseaux de neurones *feedforward* est dense dans l’espace des fonctions mesurables² définies sur et à valeurs dans un espace de dimension fini.

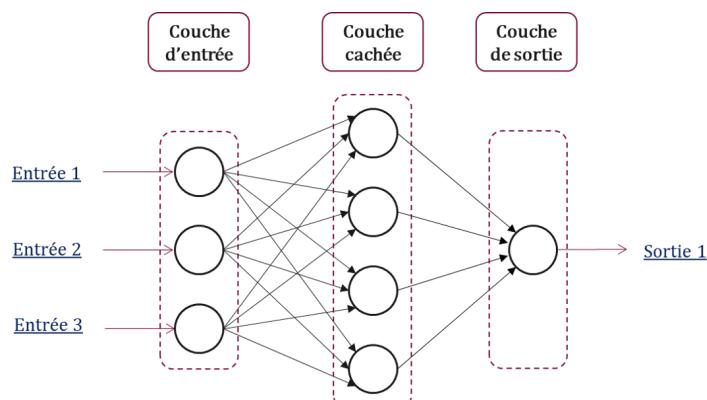


FIGURE 3.2 – Représentation d’un réseau de neurones dense *feedforward* à une seule couche cachée.

2. Au sens de la mesure de Lebesgue

La restriction à une architecture à une seule couche cachée n'est pas privilégiée car le nombre de neurones nécessaire à une bonne approximation peut être très important, ce qui a un impact sur la complexité computationnelle de l'entraînement. C'est pourquoi en pratique, on emploie des réseaux de neurones possédant plusieurs couches cachées.

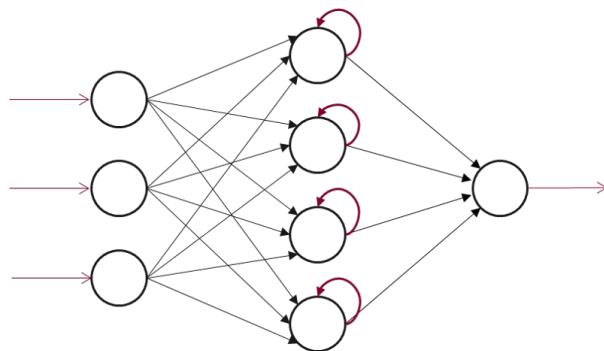
3.4 Les réseaux de neurones récurrents

Dans le cas du traitement des données séquentielles et contextuelles, une catégorie particulière de réseaux de neurones se distingue par ses aptitudes à rendre compte de la connaissance du passé : le réseau de neurones récurrent ou *RNN*, *Recurrent Neural Networks*. Ce type de réseau de neurones intervient dans l'architecture du modèle de provisionnement autour duquel nous avons travaillé ; nous consacrons cette section à une présentation. Nous expliquerons dans un premier temps la particularité des réseaux de neurones récurrents, nous soulèverons ensuite les difficultés de mémorisation des *RNN* en général, enfin nous nous intéresserons à une structure récurrente spécifique le *Gated Recurrent Unit*.

3.4.1 Caractérisation des réseaux de neurones récurrents

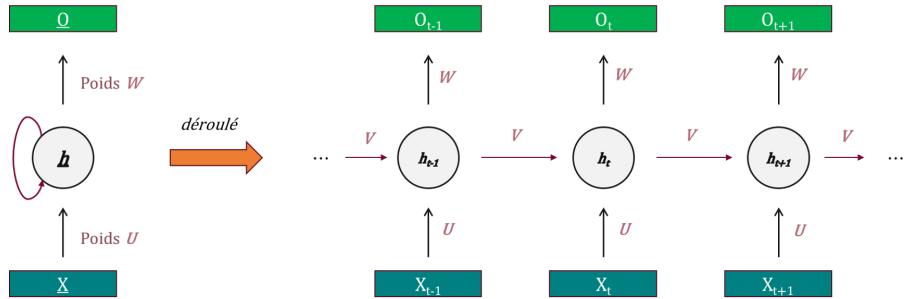
Les réseaux de neurones récurrents *RNN* sont un type de réseaux de neurones qui présentent la particularité d'être dotés de boucles de rétroaction. Ils permettent de tenir compte des situations où la prédiction à réaliser dépend non seulement des données fournies en entrée mais également des états antérieurs du neurone.

FIGURE 3.3 – Aperçu d'un réseau de neurones récurrent.



Une telle situation est équivalente à considérer un état caché, où l'on met en mémoire la valeur antérieure.

FIGURE 3.4 – Représentation d'un *RNN* déroulé.



3.4.2 La mémoire des réseaux de neurones récurrents

L'utilisation des réseaux de neurones récurrents nous confronte à un problème : au cours de la rétro-propagation on peut observer des situations d'évolution extrême du gradient. Elles résultent à la fois du partage des poids entre toutes les couches et du choix des fonctions d'activation. On assiste ainsi à des phénomènes de *gradient vanishing*, disparition du gradient, ou de *gradient exploding*. Ces phénomènes ont pour conséquence de minimiser la contribution de l'erreur mesurée en sortie à l'ajustement de la valeur des poids des couches éloignées de la sortie. Ces phénomènes empêchent le réseau récurrent standard d'apprendre des données séquentielles anciennes. Dans le cas d'une série temporelle, ce sont les observations chronologiquement éloignées ; dans le cas de données textuelles, deux mots très éloignés.

On peut illustrer le *gradient vanishing* en prenant l'exemple de la fonction sigmoïde. Cette fonction d'écrasement étant définie sur \mathbb{R} et à valeurs comprises entre 0 et 1, on observe que sa dérivée est faible puisque qu'une variation importante de l'argument n'entraîne qu'une variation très faible de l'image par la fonction sigmoïde.

Cette situation a deux impacts :

- D'abord sur la mise à jour des poids. Si le gradient est très faible alors les poids sont très peu modifiés puisque la correction apportée aux poids est proportionnelle au gradient.
- Ensuite sur la valeur du gradient pour les couches plus hautes : par application du théorème de dérivation des fonctions composées, la dérivée des fonctions d'activation des couches basses intervient de façon multiplicative dans le calcul du gradient des couches hautes. Ainsi, le gradient des couches hautes s'en trouve atténué et les données éloignées n'influent que très peu sur la correction des poids.

Ces phénomènes de disparition et d'explosion du gradient constituent la principale limite d'un *RNN* standard.

3.4.3 Le Gated Recurrent Unit

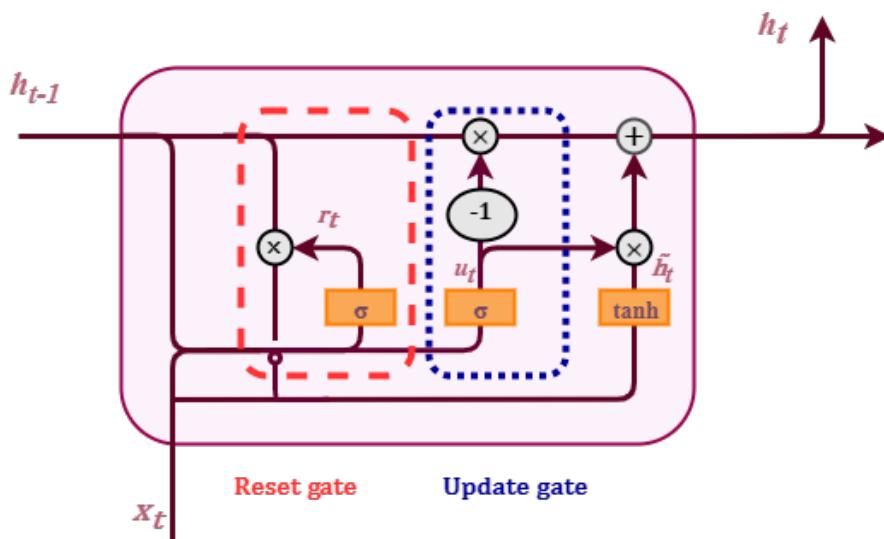
Dans le but de s'affranchir des limites du *RNN* standard que nous avons mises en évidence, des architectures spécifiques de réseaux de neurones récurrents ont été recherchées. Deux architectures en particulier sont principalement utilisées et offrent des solutions à ces problèmes : l'architecture récurrente *Long Short Term Memory LSTM* proposée pour la première fois par HOCHREITER et SCHMIDHUBER (1997), et l'unité récurrente à portes, *Gated Recurrent Unit* ou *GRU* plus récente et inspirée du *LSTM*. Dans la suite nous nous intéressons uniquement au *GRU*, qui a été employé dans la résolution de problèmes de provisionnement.

Le *Gated Recurrent Unit*, *GRU* est un réseau de neurones récurrent introduit en 2014 (CHO et al., 2014). Il présente la particularité de résoudre le problème de *gradient vanishing* et *gradient exploding* et permet ainsi d'apprendre des relations de dépendance de long terme. L'unité récurrente à portes se présente comme un bloc de construction qui peut être intégré dans un réseau de neurones récurrent plus étendu.

3.4.3.1 Fonctionnement du *GRU*

Comme son nom l'indique, l'unité récurrente à portes est composée de deux portes. Une porte de mise à jour *update gate*, et une porte de ré-initialisation *reset gate*.

FIGURE 3.5 – Schéma d'un *GRU*.



En entrée du *GRU*, on fournit :

- une entrée x_t
- l'état caché antérieur h_{t-1} , h_0 étant initialisé arbitrairement par convention.

La porte *update*

La valeur u_t est calculée comme suit :

$$u_t = \sigma(W_u \cdot x_t + W_u \cdot h_{t-1})$$

La porte *update* retourne une valeur comprise entre 0 et 1 qui traduit la proportion d'information de la couche cachée passée que l'on souhaite conserver.

Si cette valeur est proche de 0 alors l'état de mémoire actuel en sortie du *GRU* est égal à l'état précédent.

La porte *reset*

La valeur r_t est calculée comme suit :

$$r_t = \sigma(W_r \cdot x_t + W_r \cdot h_{t-1})$$

Elle permet au modèle de déterminer la part d'information à oublier et permet d'opérer un filtre quant aux informations non-pertinentes pour le futur.

On détermine ainsi une synthèse :

$$\tilde{h}_t = \tanh(W \cdot (r_t \times h_{t-1}) + W \cdot x_t)$$

Finalement, on calcul la mémoire à l'instant t h_t comme une pondération de h_{t-1} et \tilde{h}_t par la valeur u_t .

$$h_t = (1 - u_t) \times h_{t-1} + u_t \times \tilde{h}_t$$

TABLE 3.3 – Comparaison d'un RNN standard et d'un *GRU* pour la mise à jour de l'état caché *hidden state*.

$$\text{RNN standard} \quad h_t = F_W(h_{t-1}, x_t)$$

$$\begin{aligned} \text{GRU} \quad h_t &= (1 - u_t) \times h_{t-1} + u_t \times \tilde{h}_t \\ &\text{où } \tilde{h}_t = \tanh(W \cdot (r_t \times h_{t-1}) + W \cdot x_t) \end{aligned}$$

3.4.3.2 Utilisation de l'unité récurrente à portes *GRU* dans un réseau de neurones

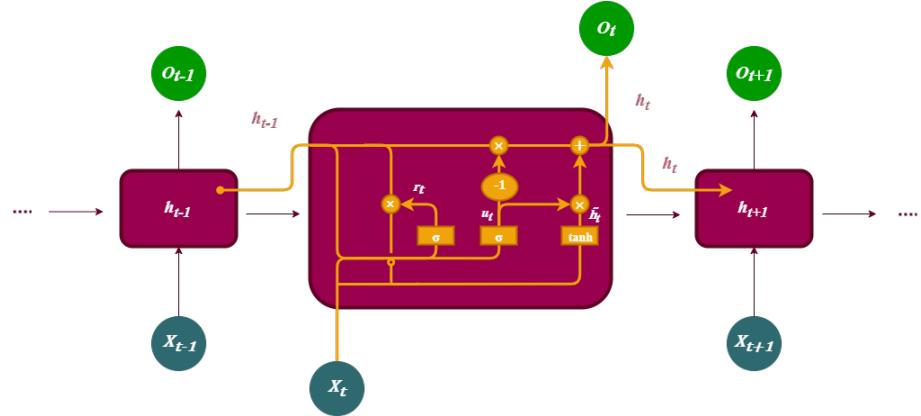
On substitue l'unité *GRU* à un neurone unique. On peut ainsi former des couches *GRU* à n^3 unités résultant de l'association de plusieurs unités *GRU*.

Leur utilisation en lieu des neurones simples permet de réaliser des structures récurrentes,

3. En raison d'un héritage historique, il est courant de choisir n comme une puissance de 2.

dont la prédition dépend à la fois de l'information d'entrée et de l'état antérieur de la mémoire et qui sont à même de traiter les informations séquentielles. Les *GRU* se distinguent des *RNNs* classiques car s'affranchissant des problèmes de *gradient vanishing* et *gradient exploding*, ils permettent de traiter des problèmes de mémoire à long terme.

FIGURE 3.6 – Schéma de la mise en cascade de plusieurs unités *GRU*.



Le *GRU* qui permet de mémoriser le passé à plus long terme et de filtrer les données non-pertinentes, constitue ainsi une structure récurrente d'intérêt dans le cadre de notre étude de l'apport de réseaux neuronaux au provisionnement IARD.

Deuxième partie

**Le modèle *Deep Triangle* de Kévin
Kuo**

Chapitre 4

Une présentation du modèle *Deep Triangle*

Dans la recherche de nouveaux modèles de provisionnement, un intérêt croissant est porté aux méthodes s'inscrivant dans le domaine du *Deep Learning* et mettant en œuvre les réseaux de neurones. Parmi les modèles publiés au cours des trois dernières années figure un modèle particulier dont nous avons décidé d'étudier les spécificités : *Deep Triangle* (KUO, 2018).

Le modèle *Deep Triangle* auquel nous avons consacré notre étude est un modèle de provisionnement qui s'appuie sur l'utilisation de réseaux de neurones dont l'architecture présente des structures récurrentes. Il propose d'entraîner un réseau de neurones récurrent afin qu'il puisse prédire conjointement le montant de paiement incrémental du sinistre ainsi que le montant de réserve dossier/dossier pour l'année de développement à venir. L'information fournie en entrée du réseau de neurones est un historique des paiements incrémentaux et des charges au cours des années de développement antérieures.

La réflexion sous-jacente au modèle *Deep Triangle* repose sur la prise en compte de l'évolution temporelle du sinistre d'une année de développement à l'autre afin de prédire les montants à venir : on utilise la connaissance des données du passé dans l'objectif de prédire les *cash-flows* futurs.

Dans l'approche *Deep Triangle* présentée par Kevin Kuo, on considère un réseau de neurones différent pour chaque secteur d'activité (*LoB*¹). Cette démarche est cohérente d'un point de vue actuariel puisque dans la réalité on observe effectivement des cadences de développement très différentes selon le secteur d'activité. Afin de pouvoir être à même de réaliser ces prédictions, le réseau de neurones doit préalablement être entraîné avec des données de sinistres. Pour un secteur d'activité donné, le réseau de neurones, une fois entraîné avec les données de sinistres, permet de réaliser des prédictions pour les sinistres de ce secteur d'activité.

La particularité du modèle *Deep Triangle* est qu'il se propose de prédire conjointement les paiements et les charges. Cette double compétence est pertinente si l'on adopte un point de vue actuariel, quand on sait que pour certains secteurs d'activité on raisonne

1. le terme *LoB* est l'acronyme de l'anglais *line of business*, il très couramment utilisé en assurance pour désigner un secteur d'activité.

plutôt sur les triangles de charges ou sur les triangles de paiements compte tenu de la nature volatile de l'activité.

4.1 Présentation du problème à résoudre

Pour un agrégat de sinistres rattaché à une année de survenance i donnée, on cherche à prédire le montant du paiement incrémental et le montant de la réserve pour l'année de développement j à venir, connaissant l'historique de ces quantités pour les T dernières années de développement. La valeur de la durée de l'historique est un paramètre. Cette tâche de prédiction à un an, peut être réitérée en cascade pour plusieurs années de développement afin d'aboutir à une prédiction des montants à l'ultime N . Nous avons réalisé notre étude en choisissant la valeur $T = N-1$. Les données dont nous disposons nous permettent de procéder à la résolution de notre problème de provisionnement par apprentissage supervisé.

4.1.1 Les variables explicatives utilisées

Les informations à fournir en entrée du réseau de neurones sont de trois natures : elles correspondent aux montants de paiement antérieurs, aux montants de réserve antérieurs, et à l'identifiant de l'entreprise.

Pour chaque secteur d'activité et pour une entreprise donnée on s'intéresse à des sinistres survenus au cours d'une période de $I=N=10$ ans. Pour un agrégat de sinistres survenus en l'année $i \in [1, I]$, on dispose pour chaque année de développement de $(C_{i,j})_{1 \leq j \leq J=10}$, l'ensemble des paiements incrémentaux, et $(OS_{i,j})_{1 \leq j \leq J=10}$, les montants totaux de charges non payées (réserves dossier/dossier).

Pour des raisons de performance, il est plus pertinent dans le cas de notre problème de travailler avec des données qui sont du même ordre de grandeur ainsi que le mettent en lumière LECUN et al. (1998) dans le chapitre *Efficient BackProp* de leur ouvrage commun. Aussi on procède à une normalisation de ces quantités en utilisant le montant de primes acquises en l'année i . Pour une année de développement k , le couple des ratios ainsi obtenu est noté

$$Y_{i,k} = \left[\frac{C_{i,k}}{NPE_i}, \frac{OS_{i,k}}{NPE_i} \right]$$

Cette normalisation est en conformité avec la proposition de KUO, 2018.

Le modèle rend possible l'inclusion des données de plusieurs entreprises dans l'échantillon d'entraînement. On considère les données d'historique de sinistres d'un ensemble de \mathcal{C} entreprises différentes $\{c \in \mathcal{C}\}$. Les codes identifiant les entreprises sont ramenés à des valeurs catégorielles numérotées entre 0 et $|\mathcal{C}| - 1$

4.1.2 Une formalisation de la tâche d'apprentissage

Notre tâche d'apprentissage supervisé est telle que l'information d'entrée X contient deux composantes :

- x_1 , de dimension $(T, 2)$, la série temporelle historique des T années précédant l'année de développement à prédire
- x_2 , de dimension $(1, 1)$, un entier comprise entre 0 et $|\mathcal{C}| - 1$ qui correspond à un code `id_group_code` identifiant l'entreprise.

$$X = [x_1, x_2]$$

$$X = [[Y_{i,j-1}, Y_{i,j-2}, \dots, Y_{i,j-8}, Y_{i,j-T}] , \text{ id_group_code }]$$

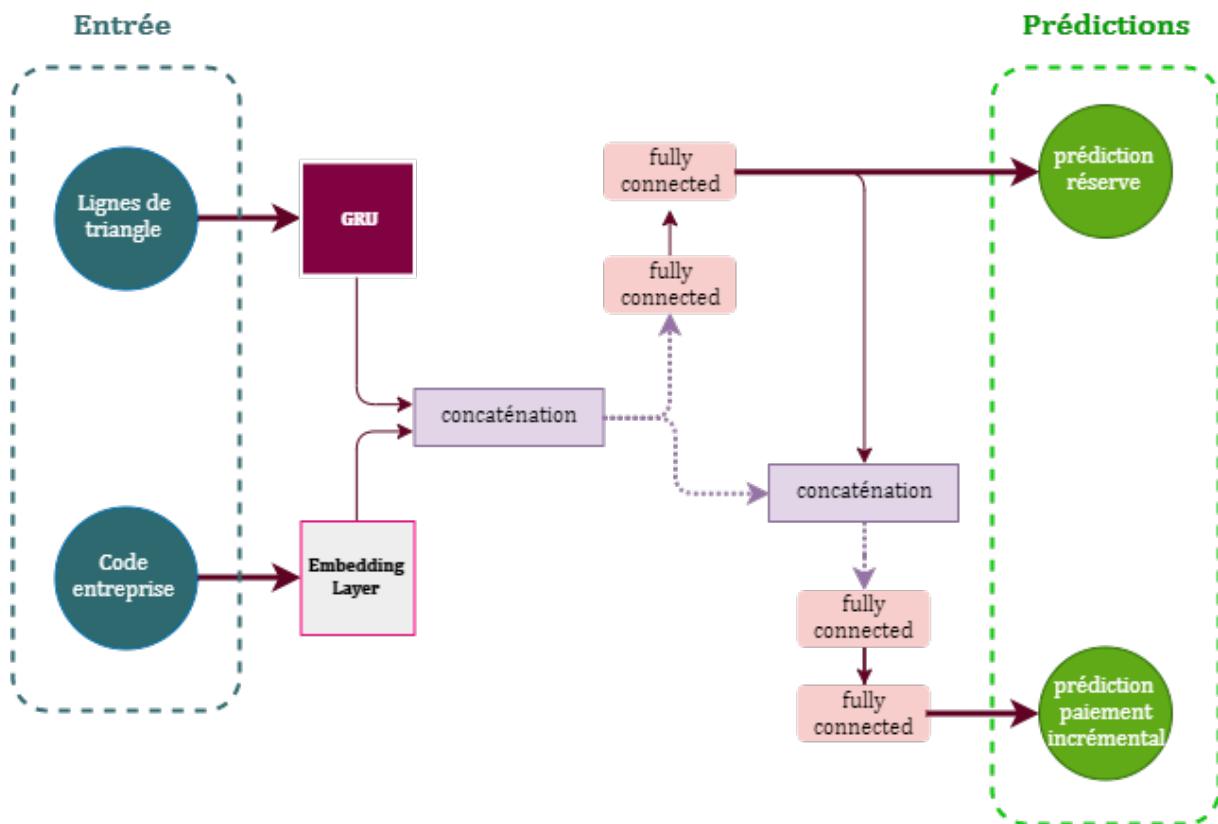
L'information de sortie Y de dimension $(2, 1)$ est un vecteur dont les composantes sont la prédition du montant de paiement incrémental et la prédition de la réserve dossier/dossier pour l'année de développement ultérieure j telle que

$$\begin{aligned} Y &= Y_{i,j} \\ &= \left[\frac{C_{i,j}}{NPE_i}, \frac{OS_{i,j}}{NPE_i} \right] \end{aligned}$$

4.2 L'architecture du réseau de neurones

4.2.1 Aperçu de l'architecture

FIGURE 4.1 – Architecture du réseau de neurones *DeepTriangle*.



L'architecture du modèle *Deep Triangle* se présente comme une structure faisant intervenir une couche *GRU*, une couche *embedding*, plusieurs couches de concaténation des résultats, ainsi que des couches *fully connected*, qui correspondent à des couches dont tous les neurones sont connectés à tous les neurones des couches antérieures et suivantes. La fonction d'activation pour les couches *fully connected* est une fonction sigmoïde. L'information sur les montants est traitée par la couche *GRU* et l'identifiant de l'entreprise par la couche *embedding*. Les deux résultats sont ensuite concaténés et traités par des couches *fully connected* afin d'obtenir en sortie les montants de paiement incrémentaux et de réserve dossier/dossier.

4.2.2 Une couche *embedding*

De façon générale, une couche *embedding* permet de convertir les modalités d'une variable catégorielle en vecteurs continus dans un espace de dimension finie de notre

choix. La transformation réalisée n'est pas quelconque : elle permet dans le même temps de capturer des informations quant aux relations entre les différentes modalités de la variable catégorielle. Le recours à la couche *embedding* permet donc d'obtenir en sortie des représentations vectorielles caractérisant de façon unique chaque modalité et qui sont aussi porteuses d'informations quant au lien de proximité entre ces modalités.

Dans le modèle *Deep Triangle*, la couche *embedding* prend en entrée les codes identifiant les entreprises et l'apprentissage des liens de proximité est réalisé au cours de l'entraînement du réseau de neurones.

Dans l'espace d'*embedding*, la proximité de deux vecteurs au sens de la distance euclidienne est représentative de la proximité des entreprises qu'ils représentent pour la tâche de provisionnement à laquelle on s'intéresse. Cette caractéristique informative permet de mettre en évidence la différence avec une autre méthode d'encodage assez populaire en *machine learning* qui est l'encodage *one-hot* qui consiste en la représentation d'une modalité par un vecteur de booléens. Pour chaque ligne de business, on s'intéresse aux informations de $C = 50$ entreprises. La couche *embedding* convertit alors les 50 identifiants uniques des entreprises en 50 vecteurs de dimension 49.

4.2.3 Un bloc *GRU*

L'architecture *Deep Triangle* inclus une couche associant des unités *GRUs*, dont nous avons présenté précédemment le fonctionnement.

Cette couche *GRU* est composée de 64 unités et intervient dans l'analyse des séries temporelles bi-variées que nous fournissons en entrée.

4.3 Paramètres d'entraînement du modèle

4.3.1 Fonction de coût

La fonction de coût proposée pour l'entraînement du réseau *Deep Triangle* est l'erreur absolue moyenne *MAE*. La fonction de coût du modèle global s'exprime comme une pondération des deux fonctions de coût, à savoir la *MAE* pour la prédiction des paiements incrémentaux et de la *MAE* pour la prédiction des réserves. La pondération des contributions de chacun des écarts de prédiction à la fonction de coût du modèle global sont des paramètres qui peuvent être choisis. Dans l'approche présentée par KUO, 2018, les valeurs de poids proposées sont 0.2 pour l'erreur de prédiction des réserves et 0.8 pour l'erreur de prédiction des paiements. Le choix de ces valeurs traduit une prépondérance de l'erreur de prédiction des paiements dans la détermination des poids optimaux du modèle.

$$MAE_{model} = 0,8 * MAE_C + 0,2 * MAE_{OS}$$

Nous avons retenu la même pondération afin de pouvoir procéder à une analyse critique de l'approche de Kevin Kuo.

4.3.2 Algorithme d'optimisation *AMSGrad*

La méthode de descente de gradient possède plusieurs variantes. Chacune de ces variantes se propose d'apporter à l'algorithme d'optimisation des caractéristiques permettant de faciliter la convergence vers un voisinage de l'optimum. L'intérêt des différentes variantes réside dans leur objectif de proposer une méthode pour se prémunir d'une limite de l'algorithme de descente de gradient général. Par exemple, la descente de gradient stochastique, qui considère un *batch* de données, permet de s'adapter à un volume de données important.

L'algorithme d'optimisation proposé par Kevin Kuo pour l'entraînement du modèle *Deep Triangle* est l'algorithme *AMSGrad*, présenté en 2018 par REDDI et al. (2018). Il se présente comme une variante de la méthode d' "estimation adaptative du moment" connue sous le nom *Adam* et introduite par KINGMA et LEI BA (2015).

La méthode *Adam* est une méthode de descente de gradient spécifiquement destinée à l'entraînement des réseaux neuronaux. Le cœur de son fonctionnement est qu'elle permet de déterminer certains paramètres automatiquement et de les mettre à jour au cours des itérations, là où une valeur arbitrairement renseignée et fixée au cours de l'entraînement pourrait ne pas assurer des conditions de convergence optimales.

La mise en œuvre de cette méthode nous place dans le cas de la résolution de programme d'optimisation où la valeur du paramètre *learning rate* est déterminée par l'algorithme et se met à jour à chaque itération selon les valeurs antécédentes des gradients. Comme nous l'avons exposé précédemment, le *learning rate* est un paramètre important dont la valeur est susceptible d'influer très fortement sur la qualité de l'entraînement. Ainsi, une méthode adaptative telle qu'*Adam* est particulièrement indiquée pour l'entraînement des réseaux de neurones et offre un avantage du point de vue de la convergence de la méthode de descente de gradient. De plus, la méthode *Adam* propose également de contrôler les phénomènes de décroissance du *learning rate* que l'on peut observer empiriquement avec d'autres algorithmes adaptatifs.

La méthode *AMSGrad* hérite de toutes les propriétés mélioratives d'*Adam* tout en permettant de se prémunir des cas de non-convergence de l'algorithme *Adam* mis en lumière par les recherches de REDDI et al., 2018.

Ainsi, à l'issue d'une confrontation de la proposition faite par KUO, 2018 avec l'état actuel de l'art, il nous a paru pertinent de retenir l'algorithme *AMSGrad* pour entraîner le modèle que nous avons implémenté.

Chapitre 5

Une exploration des données utilisées

5.1 Une description statistique des données de la base d'étude

Les données que nous avons utilisées pour entraîner notre modèle et en mesurer les performances sont issues de la base de données *Schédule P* de la NAIC¹, l'association nationale des commissaires d'assurance, organisme américain de normalisation et d'appui à la réglementation assurantielle.

La base de données rassemble des sinistres survenus sur une période de 10 ans entre 1988 et 1997 ainsi que les données concernant leur évolution au cours d'une période de 10 années après leur survenance. Les sinistres contenus dans la base se rapportent à une multitude d'entreprises et plusieurs secteurs d'activité. Pour chaque secteur d'activité, les sinistres d'une même entreprise sont agrégés selon leur année de survenance. Un agrégat est ainsi caractérisé par un secteur d'activité, un assureur et une année de survenance.

Dans un soucis de rigueur nous avons procédé à notre étude en considérant le même échantillon de données que Kevin Kuo : il s'agit d'un échantillon de la base *Schedule P*, regroupant 4 secteurs d'activité et 50 entreprises pour chacun des secteurs d'activité. Nous l'appellerons base de données d'étude dans la suite.

5.1.1 Dictionnaire des variables

La base de données d'étude rassemble l'évolution chronologique des agrégats de sinistres, qui sont décrits par les variables présentées ci-après. Notre base de données d'étude comporte 20 000 entrées. Chaque entrée de la base de données correspond à la vision à une année de développement donnée j , d'un agrégat de sinistres survenus au cours d'une année i . On peut donc faire correspondre chaque ligne de la base de données d'étude à une cellule d'un triangle liquidation.

1. NAIC : National Association of Insurance Commissioners.

TABLE 5.1 – Présentation des variables décrivant une entrée de notre base de données.

Variable	Description	Type
LoB	Secteur d'activité	qualitative nominale
$group_code$	Code identifiant le groupe	qualitative nominale
$group_name$	Nom du groupe	qualitative nominale
i	Année de survenance des sinistres de l'agrégat	quantitative discrète
j	Année de développement considérée	quantitative discrète
$C_{i,j}$	Montant du paiement incrémental	quantitative continue
$Cum_{i,j}$	Montant du paiement cumulé	quantitative continue
$Inc_{i,j}$	Montant de la charge	quantitative continue
$OS_{i,j}$	Montant de la réserve, différence entre les charges et les payés	quantitative continue
NPE_i	Volume de primes acquises net pour l'année de survenance i des sinistres de l'agrégat.	quantitative continue

5.1.2 L'activité des individus de notre base de données

Les données utilisées pour l'entraînement du modèle *Deep Triangle* couvrent 4 secteurs d'activité :

- ***commercial auto/truck liability/medical*** : couverture pour tous les véhicules destinés à un usage professionnel, qu'il s'agisse de voitures de fonction, d'utilitaires ou d'outils de travail. Elle couvre notamment contre les dommages aux biens et inclut la responsabilité civile².
- ***private passenger auto liability/medical*** : assurance automobile, pour particuliers.
- ***workers'compensation*** : assurance accident du travail. A l'initiative de l'employeur, au bénéfice des employés en cas d'accident du travail ou de développement d'une maladie professionnelle. Aux États-Unis la prestation s'inscrit parmi différentes catégories telles que : indemnité de remplacement de revenu, soins médicaux, ou encore réadaptation professionnelle.
- ***other liability*** : assurance responsabilité. Elle inclut les protections contre la responsabilité légale résultant de la négligence, ou de l'inaction et ayant causé des dommages matériels ou des blessures à autrui.

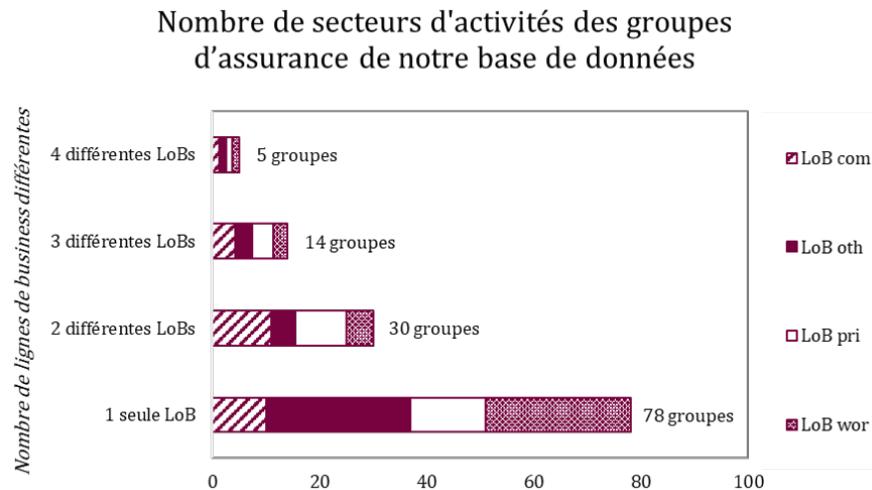
Lorsque nécessaire, nous employons les trois premières lettres du nom de la LoB pour nous y référer. Les quatre $LoBs$ sont ainsi identifiées par les notations *com*, *oth*, *pri* et *wor*.

Chacun de ces quatre secteurs d'activité regroupe exactement 50 groupes différents. Un même groupe pouvant être présent sur de multiples secteurs d'activité, on dénombre au total 127 groupes d'assurance différents dans la base de données d'étude. Une entreprise peut ainsi être définie de façon unique par la donnée conjointe du groupe auquel elle

2. On parle aux US de *Property & Casualty*.

appartient et du secteur d'activité concerné.

FIGURE 5.1 – Représentation de la multiplicité des activités des groupes d'assurance de notre base de données d'étude.



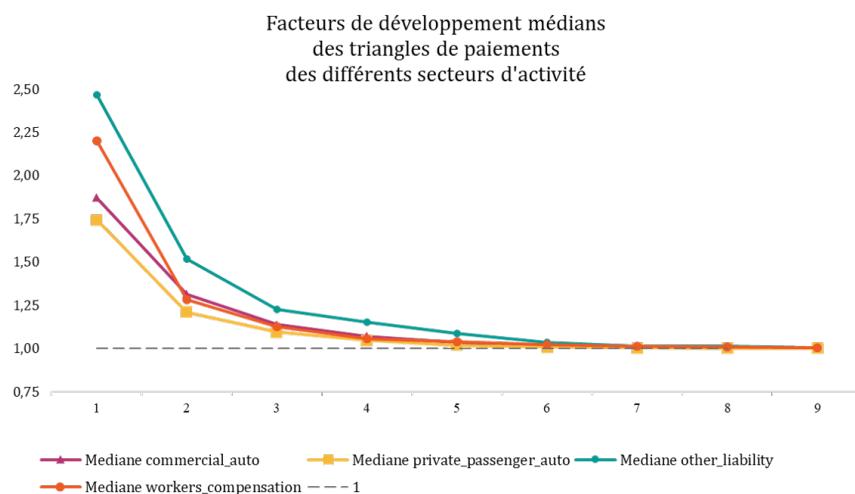
Parmi les 127 groupes, seuls 5 exercent sur les quatre LoBs que comporte notre base d'étude. 78 groupes ne sont présents que sur l'une des quatre LoBs.

Ainsi 61% des groupes de notre base d'étude n'exercent d'activité que sur l'une des quatre LoBs. Dans ce cas, elles exercent en majorité dans les secteurs d'activité *workers' compensation* ou *other liability*.

5.1.3 Le profil et les flux financiers de nos entreprises

5.1.3.1 Les facteurs de développements sur chacune des *LoBs*

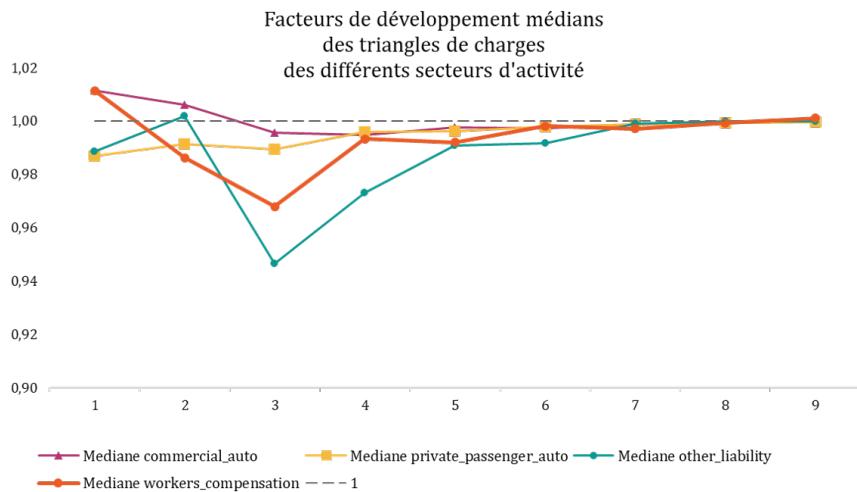
FIGURE 5.2 – Comparaison de l'évolution des coefficients de Mack-Chain-Ladder de chacun des secteurs d'activité : triangles de paiements.



Pour la *LoB workers' compensation* le facteur médian de Chain Ladder de l'année 1 vaut 2,25. Ce facteur est égal à la médiane des facteurs de développement de Chain-Ladder de l'année 1 pour l'ensemble des entreprises de la *LoB*.

La limite des facteurs de Chain-Ladder vaut 1. Une stabilisation des coefficients à cette valeur correspond à la clôture du sinistre. Ainsi, la représentation des coefficients médians de Chain-Ladder nous permet de comparer la vitesse de convergence de la suite des facteurs médians vers la valeur 1. Cette vitesse nous permet de caractériser la cadence de règlement des sinistres de chaque *LoB*. On identifie ainsi la *LoB other liability* comme une *LoB* à développement long, et la *LoB private passenger auto* comme une *LoB* à développement court.

FIGURE 5.3 – Comparaison de l'évolution des coefficients de Mack-Chain-Ladder de chacun des secteurs d'activité : triangles de charges.

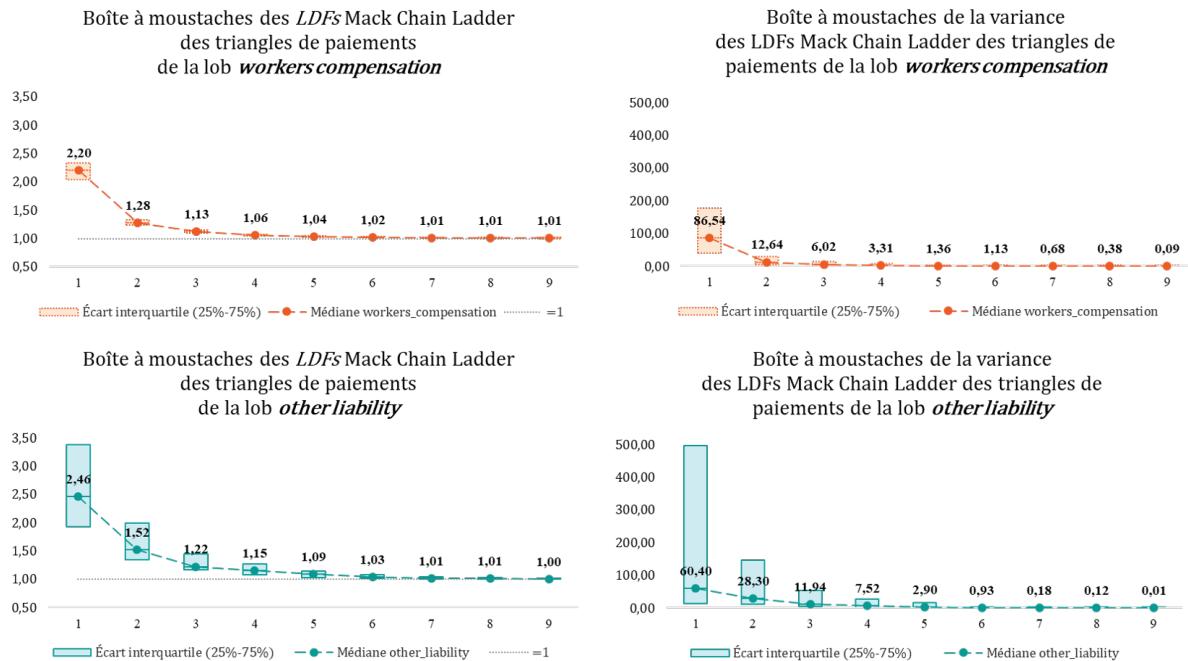


Pour les triangles de charges de la LoB private passenger auto, le facteur de développement médian pour l'année de développement 3 vaut 0,95.

Les facteurs de développement médians des triangles de charges mettent en évidence les différentes typologies d'activité. Ainsi une évolution très importante des charges anticipées par le gestionnaire de sinistres traduit une plus grande volatilité de la LoB.

Le modèle de Mack-Chain-Ladder nous permet de calculer la variance de Mack des facteurs de développement. Nous nous intéressons à ces grandeurs pour deux *LoBs*, les autres graphes sont disponibles en annexe (Figure A.2 et Figure A.3). Nous considérons l'évolution chronologique des boîtes à moustaches pour chaque année de développement.

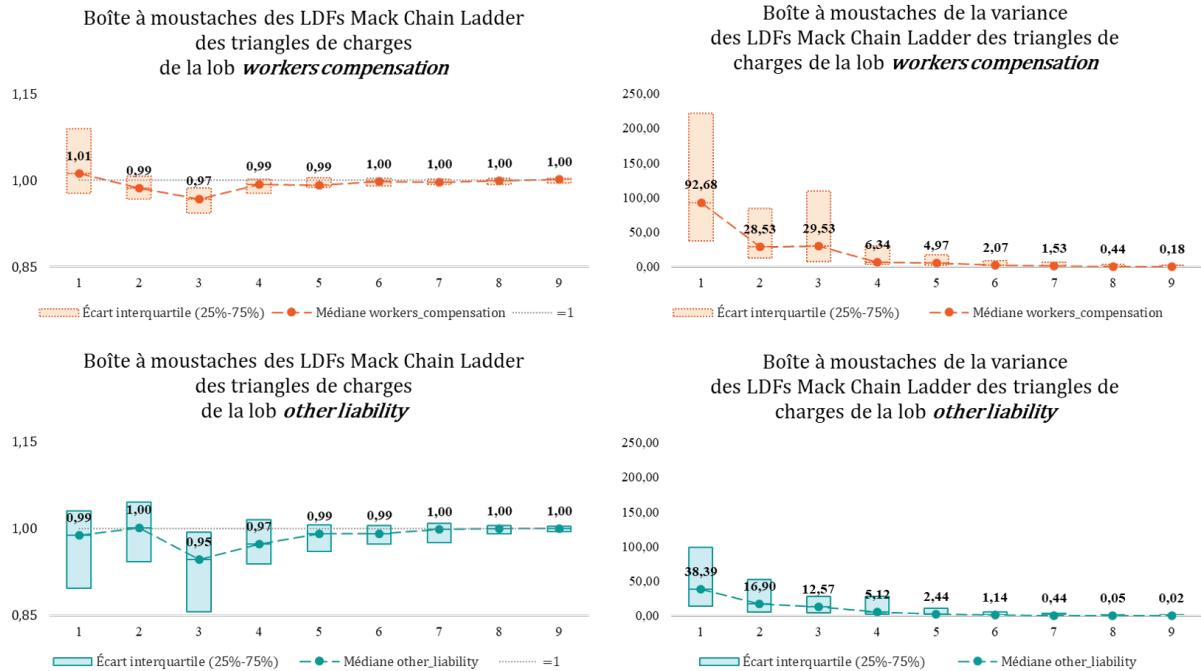
FIGURE 5.4 – Comparaison des caractéristiques de deux *LoBs* : triangles de paiements.



Pour la *LoB workers compensation* on observe un resserrement de la distribution des variances des facteurs de développement autour de la valeur 0 et une décroissante très rapide de la variance médiane de 86,54 à 0,09 entre les années de développement 1 et 9.

Pour la *LoB other liability* le déroulement est moins rapide et les variances sont plus élevées au cours des 5 premières années de développement.

FIGURE 5.5 – Comparaison des caractéristiques de deux *LoBs* : triangles de charges.

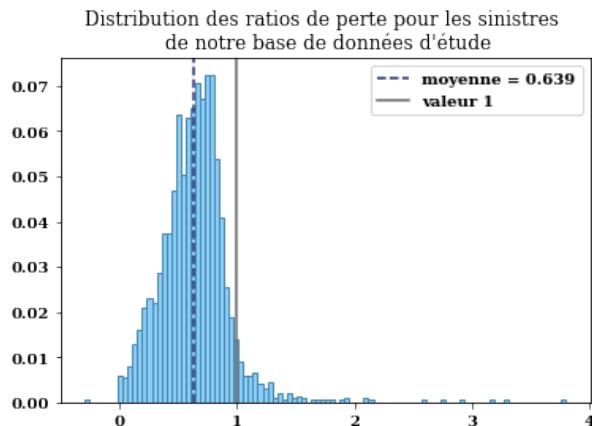


L'étude des quatre *LoBs* met ainsi en évidence le caractère plus volatile de la *LoB other liability* à développement long.

5.1.3.2 Les ratios de perte

Afin de pouvoir mesurer la sinistralité des entreprises de notre base de données d'étude à l'aune du volume de primes qu'elles perçoivent, nous nous sommes intéressée aux ratios de perte. Pour chaque entreprise, le ratio de perte S/P est le rapport des paiements à l'ultime et des primes encaissées au cours de l'année de survenance.

FIGURE 5.6 – Analyse des ratios de perte des entreprises de notre base de données d'étude.



Nous avons représenté la distribution des ratios de perte pour l'ensemble des groupes de notre base de données. La distribution empirique des ratios de perte est centrée autour de la valeur 0,639.

Nous présentons en annexe (Figure A.1) une distinction par secteur d'activité des ratios de perte. On observe que le ratio de perte hors frais divers est en moyenne égal à 64% (brut de frais).

5.1.3.3 Poids des entreprises dans chaque *LoB* selon leur volume de primes

FIGURE 5.7 – Représentation graphique de la part des primes des assureurs les plus importants, au seuil de 80% de part de primes cumulées.



Nous nous intéressons à la part des primes de chaque assureur sur une *LoB* pour pouvoir comparer les entreprises de notre base de données. Le volume de primes considéré est le volume de primes total sur la période temporelle de 1988 à 1997. On représente l'ensemble des acteurs les plus importants au seuil de 80%.³

Un faible nombre d'assureurs différents appartenant à cet ensemble traduit la concentration du marché. Ainsi pour la *LoB private passenger auto*, deux entreprises représentent à elles seules 89% du volume des primes de la *LoB*, tandis que les 48 autres que recense la base de données d'étude pèsent ensemble pour seulement 11% du volume de primes.

Nous observons ainsi que les acteurs présents dans notre base de données sont d'importances différentes. Or, le modèle de provisionnement que nous avons développé se propose

3. Il en vient qu'au seuil de 100% tous les assureurs de notre base de données seraient inclus.

de considérer un unique réseau de neurones pour l'ensemble d'une *LoB*. Cette situation nous a poussé à nous interroger sur les choix à faire quant à l'usage des données afin de tenir compte des importances inégales de nos acteurs.

Afin de pouvoir proposer un modèle aussi bien adapté aux acteurs majeurs qu'aux assureurs de plus petite envergure, nous avons fait le choix d'attribuer le même poids à chaque entreprise et de ne pas sur-pondérer les observations d'une entreprise qui représenterait une part importante du marché. On pourrait s'inquiéter de ce que l'attribution de la même pondération à un petit acteur et un acteur majeur puisse avoir pour effet une contamination des prédictions pour les acteurs prépondérants par les résultats des assureurs plus petits. Toutefois, l'architecture de notre modèle nous permet, par l'application de la couche *embedding*, de traduire la proximité de deux entreprises lors de la conversion de leur identifiant en vecteur spatial. Les poids de la couche *embedding* sont mis à jour par rétropropagation des erreurs des couches plus basses ayant pour tâche de prédire les montants futurs, et déterminent entièrement la représentation vectorielle d'une entreprise. Ainsi, un rapprochement de deux entreprises similaires et un éloignement de deux acteurs très différents est effectué avant de prédire les montants.

De ce fait, la couche *embedding* pourrait être considérée par son action comme une protection qui permettrait d'attribuer le même poids à des acteurs d'envergures inégales pendant la phase d'entraînement, tout en n'entachant pas le potentiel d'exploitation du modèle pour différents types d'acteurs.

Chapitre 6

Notre implémentation du modèle

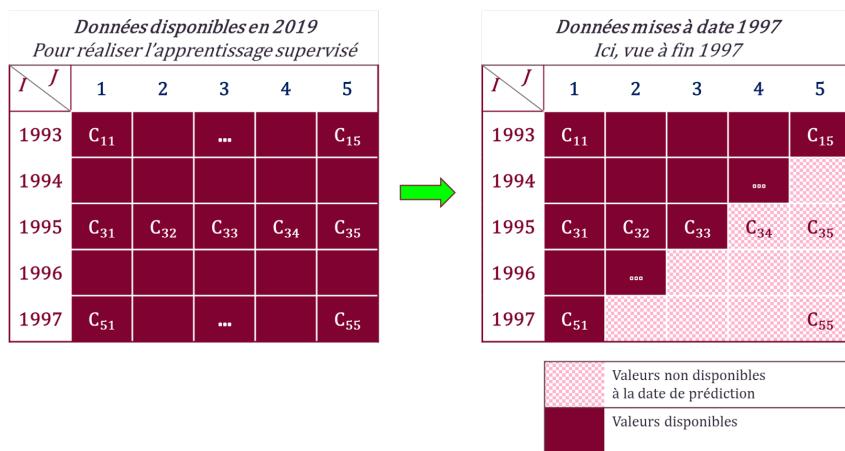
6.1 Utilisation des données pour la construction du modèle

6.1.1 Constitution des échantillons d'entraînement et de test

Conservation de l'information connue à la date de prédiction

La mise en œuvre de l'apprentissage supervisé nécessite de disposer d'un échantillon étiqueté, c'est-à-dire de plusieurs triangles de développement complétés. Toutefois, dans un souci de pertinence, il faut veiller à ne pas inclure les valeurs que l'on cherche à prédire dans notre échantillon d'entraînement. On opère donc une mise à date de nos données : on choisit une date et on ne conserve que les informations connues à cette date pour réaliser l'entraînement, les autres informations étant masquées.

FIGURE 6.1 – Illustration de la mise à date des données.



		Données disponibles en 2019 Pour réaliser l'apprentissage supervisé				
I	J	1	2	3	4	5
1993		C ₁₁		...		C ₁₅
1994						
1995		C ₃₁	C ₃₂	C ₃₃	C ₃₄	C ₃₅
1996						
1997		C ₅₁		...		C ₅₅

		Données mises à date 1997 Ici, vue à fin 1997				
I	J	1	2	3	4	5
1993		C ₁₁				C ₁₅
1994					...	
1995		C ₃₁	C ₃₂	C ₃₃	C ₃₄	C ₃₅
1996			...			
1997		C ₅₁				C ₅₅

Valeurs non disponibles à la date de prédiction
 Valeurs disponibles

Représentation de la mise à date dans le cas où $J=5$.

Cette séparation des données est en adéquation avec une approche métier, pour laquelle on dispose en pratique de données disponibles à une date précise, que l'on emploie en vue de prédire les flux de trésorerie futurs.

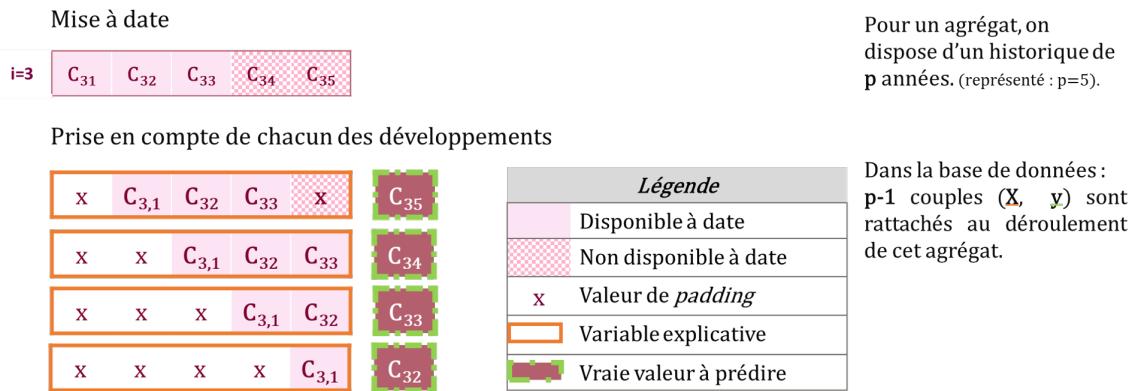
Séparation des informations d'entraînement et de test sur la base chronologique.

La mise à date que nous avons exposée consiste à considérer les données disponibles en 1997 dans notre base de données d'entraînement, et les données n'étant pas encore disponibles en 1997 dans notre base de test. Cette deuxième catégorie correspond aux sinistres employés uniquement pour réaliser des prédictions avec le modèle entraîné.

6.1.2 Emploi de chacun des développements des sinistres

Chaque développement du sinistre d'une année à l'autre est une information utile et employée pour entraîner notre réseau de neurones.

FIGURE 6.2 – Utilisation des données de sinistres pour la réalisation de l'entraînement.



Sur cet exemple nous représentons le cas où $J=5$ et mettons en lumière comment un historique de durée 5 ans donne lieu à l'ajout de quatre éléments distincts dans la base d'entraînement.

Étant donné un sinistre survenu au cours d'une année i et dont on dispose du développement sur $J = 10$ années, on peut construire $J - 1 = 9$ couples (X, y) qui permettront d'entraîner notre réseau de neurones à prédire la valeur de développement suivante. Pour ce faire, on peut donner en information d'entrée tous les couples (X, y) tels que X est l'information disponible à la fin de chaque année de développement $j \in [1, (J - 1) = 9]$ et y est la valeur pour l'année de développement $j + 1$.

Gestion de la contrainte de dimension des données

Pour un sinistre dont on cherche à prédire les paiements incrémentaux et la réserve en année de développement $j \leq J - 1$, on dispose seulement d'un historique de $j - 1 < J - 1$ ans. La taille de l'historique à fournir en entrée étant fixée, on a recours à une valeur de *padding* que l'on peut choisir librement. La valeur de *padding* nous permet de compléter le vecteur de l'historique avec une valeur choisie arbitrairement dans le cas d'un historique court.

Notre modèle est à même de comprendre et de prendre en charge ces situations. Pour cela on peut choisir une valeur de *masking*, un entier qui sera lu par les couches du réseau de neurones comme une valeur à ignorer traduisant une donnée séquentielle indisponible. La valeur de *masking* doit alors être égale à la valeur de *padding*.

Ayant décrit précisément l'utilisation des données, nous présentons dans la section suivante le processus d'entraînement de notre modèle. Nous avons choisi de procéder à l'implémentation de notre modèle en langage python et d'utiliser la bibliothèque Keras (CHOLLET et al., 2015), pour la construction et l'entraînement de nos réseaux de neurones.

6.2 Description des différentes phases du processus d'entraînement du réseau de neurones

La phase d'entraînement du réseau de neurones constitue l'étape cruciale de calibrage du modèle. L'objectif de cette étape est de déterminer les valeurs optimales des hyperparamètres conduisant à des prédictions optimales, en ce qu'elles minimisent la fonction de coût. Dans l'implémentation de notre modèle, on procède à un entraînement en deux étapes, conformément à la proposition de Kevin Kuo.

Dans un premier temps, on procède à un entraînement afin de déterminer la valeur optimale du nombre d'époques. Pour cela, on fait intervenir une condition d'arrêt afin de nous permettre de relever cette valeur.

Dans la seconde étape d'entraînement, on cherche à déterminer la valeur optimale des poids. On effectue pour cela un nouvel entraînement pour lequel on emploie la valeur obtenue à l'issue de la première étape comme nombre d'époques à réaliser.

6.2.1 Étape 1 : détermination du nombre optimal d'époques

Pour cette étape on sépare notre échantillon d'entraînement selon une date pivot, dans notre cas 1995. Les données qui étaient déjà disponibles à la fin de l'année 1995 vont servir de données d'entraînement, tandis que les données devenues disponibles entre 1995 et 1997 sont intégrées à un échantillon de validation, qui nous permet de contrôler d'éventuels phénomènes de sur-apprentissage¹. Pour cette première étape on considère ainsi l'échantillon d'entraînement `s1_X_train`, `s1_Y_train` et l'échantillon de validation `s1_X_validation`, `s1_Y_validation`.

Munis de ces données, on procède à l'entraînement du réseau de neurones en vue de déterminer le nombre d'époques optimal. On construit un réseau de neurones possédant

1. Le sur-apprentissage correspond à la situation où un modèle de *Machine Learning* tend à "mémoriser" les réponses. Il atteint ainsi d'excellentes performances sur les données d'entraînement qu'il a déjà vues mais peine à généraliser des prédictions pour de nouveaux éléments et perd ses pouvoirs prédictifs.

l'architecture adaptée (sous-section 4.2.1), et dont les poids sont initialisés aléatoirement. On fournit en entrée les données d'entraînement, et on choisit un nombre très élevé d'époques à réaliser (la valeur 1000 convient).

On se dote d'une heuristique d'interruption de l'entraînement. Un arrêt anticipé est déclenché dès que les deux conditions suivantes sont vérifiées simultanément pour une époque :

- la mesure de perte sur l'échantillon d'entraînement est inférieure à la mesure de perte sur l'échantillon de validation

$$\text{training loss} \leq \text{validation loss}$$

- sur l'échantillon de validation, aucune amélioration de la mesure de perte n'a été observée pendant 100 époques.

Ainsi, en notant $stop$ l'époque de déclenchement de la condition d'arrêt anticipé, le nombre d'époques réalisées est $m = \min(1000, stop)$.

Pour toutes les époques réalisées, on enregistre la moyenne mobile sur une plage de largeur 10 pour la mesure de perte de l'échantillon de validation.

$$\forall q \in \llbracket 1 ; m \rrbracket, \quad MA^{10}(\mathcal{L})_q = \begin{cases} \frac{1}{10} \sum_{k=0}^9 \mathcal{L}_{q-k} & \text{si } q \geq 10 \\ \frac{1}{q} \sum_{k=0}^{q-1} \mathcal{L}_{q-k} & \text{sinon.} \end{cases}$$

Le nombre optimal d'époques est celui qui minimise cette quantité :

$$q^* = \underset{q \in \llbracket 1 ; m \rrbracket}{\operatorname{argmin}} \quad MA^{10}(\mathcal{L})_q$$

Il est pertinent de noter que les conditions qui composent le critère d'arrêt anticipé nous permettent de contrôler un éventuel phénomène de sur-apprentissage, celui-ci se traduisant en pratique par une erreur de prédiction décroissante sur l'échantillon d'entraînement face à une erreur sur l'échantillon de validation qui augmente.

6.2.2 Etape 2 : détermination de la valeur optimale des poids du réseau

On considère pour cette deuxième phase un échantillon d'entraînement composé des données disponibles en 1997, et on n'utilise pas d'échantillon de validation car on ne cherche plus à vérifier l'adéquation des hyperparamètres du modèle.

On génère alors un nouveau réseau de neurones avec des poids initialisés aléatoirement. On communique ensuite les données d'entraînement, et on fixe le nombre d'époques à

réaliser à q^* , nombre optimal d'époques retenu à l'issue de l'étape 1.

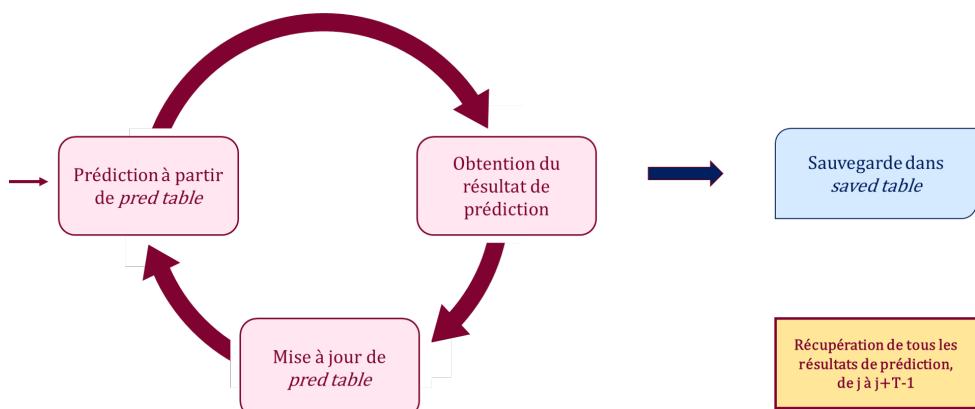
A l'issu de cette seconde phase d'entraînement, le réseau de neurones est entraîné. Les poids ont été mis à jour à chaque étape de l'entraînement afin de minimiser l'erreur de prédiction sur notre échantillon d'entraînement. On peut désormais utiliser le modèle entraîné pour réaliser des prédictions à un an. L'échantillon de test que nous avons conservé séparément peut nous permettre de vérifier les performances prédictives de notre modèle.

6.3 Utilisation du réseau de neurones entraîné pour la prédiction : conception d'un schéma itératif

Le modèle *Deep Triangle* proposé par Kevin Kuo, prend en argument un historique sur les T dernières années de développement et permet de réaliser une prédiction unique de montants pour l'année de développement suivante.

Afin de pouvoir utiliser le modèle dans le but de calculer les montants à l'ultime, il nous a fallu penser un schéma de prédiction itérative. En effet, pour chaque nouvelle année de développement, l'historique à partir duquel on prédit doit être enrichi avec la prédiction la plus récemment réalisée, avant d'être de nouveau présenté en entrée du réseau de neurone pour réaliser une nouvelle prédiction. Nous avons donc conçu un schéma itératif intégrant la prédiction par *Deep Triangle* et l'avons implémenté.

FIGURE 6.3 – Présentation du fonctionnement du schéma itératif implémenté.



Le schéma utilise deux bases de données distinctes. Une base de données *saved_table* qui est enrichie à chaque itération par ajout de nouvelles informations de prédictions, et une base de données *pred_table* qui est mise à jour à chaque itération par suppression des plus anciennes observations et intégration des plus récentes, dans le but de refléter le nouvel historique à utiliser pour la prédiction. Pour chaque *LoB*, notre schéma itératif nous a permis de procéder à la réalisation des prédictions à l'ultime à partir du réseau de neurones préalablement entraîné pour la *LoB*.

Troisième partie

Mesures de performances et approfondissement de l'étude du comportement du modèle

Chapitre 7

Mesures de performances du modèle

Afin de mesurer les performances du modèle que nous avons implémenté, nous nous sommes intéressée aux performances de prédiction à 1 an du modèle ainsi qu'aux performances de prédiction à l'ultime. Dans chacune de ces situations, nous avons procédé par une approche comparative et jaugé les performances de notre modèle à l'aune d'une méthode de provisionnement de référence : Mack-Chain-Ladder¹.

7.1 Utilisation d'une statistique synthétisant la performance et comparaison des performances de notre modèle avec une méthode de référence

Afin de mesurer les performances du modèle pour une prédiction donnée, deux solutions sont possibles :

- La première consiste à mesurer l'écart relatif entre la valeur prédite et la valeur attendue. Cette mesure est très générale et peut s'appliquer à tout problème de prédiction d'une quantité.
- La deuxième, que nous avons retenue, est de mesurer l'écart de prédiction par rapport au montant total du sinistre. Cette mesure correspond davantage à l'intérêt métier de l'assureur pour le provisionnement.

Cette considération nous permet d'adopter une approche adaptée à l'enjeu métier qui est le calcul des *IBNR*. En faisant l'hypothèse d'un ordre de grandeur similaire pour le montant total du sinistre et le volume de primes nettes, on peut alors construire notre mesure en rapportant l'écart de prédiction au volume de primes nettes. L'étude de la distribution des ratios de perte de nos agrégats de sinistres, menée dans la phase d'exploration des données que nous avons présentée (Figure 5.6), nous a permis de confirmer que cette hypothèse est crédible.

Ainsi, dans le cas de la prédiction à l'ultime J du montant total pour un agrégat rattaché à une année i , on définit donc l'erreur :

1. Nous ferons référence à Mack-Chain-Ladder par la dénomination Chain-Ladder ou l'abréviation CL, par abus de langage, bien qu'il s'agisse en toute circonstances de Mack-Chain-Ladder.

$$\delta_{C_i}^J = \frac{\hat{C}_{i,J} - C_{i,J}}{NEP_i}$$

Ce ratio correspond à l'écart entre la prédiction du modèle et la vraie valeur, exprimé par rapport aux primes nettes perçues au cours de l'année de rattachement i . On utilise cette notation pour évaluer aussi bien la prédiction des paiements que la prédiction des réserves dossier/dossier ou encore le montant de charges prédit à l'ultime.

Disposant d'un ratio pour chaque agrégat de sinistres de notre base de données, on peut utiliser une statistique des ratios ainsi définis, pour calculer une mesure de performance globale de notre modèle. Nous avons choisi de considérer l'erreur moyenne quadratique (RMSE²) pour mesurer l'écart entre le vecteur des valeurs prédites et le vecteur des valeurs attendues. Ainsi la mesure de la performance du modèle sur un échantillon de test X_{test} de cardinal $|X_{test}|$ est calculée comme :

$$\mu = \sqrt{\frac{1}{|X_{test}|} \left(\sum_{x \in X_{test}} (\delta_x^J)^2 \right)}$$

Plutôt que de définir arbitrairement un seuil à partir duquel on pourrait décider que notre statistique indique de bonnes performances, nous avons pris le parti de comparer les performances du modèle que nous avons implémenté avec une méthode de référence, la méthode de Mack-Chain-Ladder. Afin d'obtenir les résultats de prédiction par Mack-Chain-Ladder pour l'ensemble des sinistres de notre base de données, nous avons créé un module Python introduisant une classe d'objet *Triangles*, dont l'utilisation nous a permis de réaliser la prédiction des montants futurs.

Pour chacune de ces méthodes de prédiction, on s'intéresse à la prédiction à l'ultime du montant à provisionner. Il est toutefois pertinent de considérer la même statistique de performance pour la tâche de prédiction à horizon 1 an, puisque cette tâche constitue l'étape élémentaire du schéma de prédiction itératif de notre modèle.

7.2 Performances pour la prédiction à 1 an

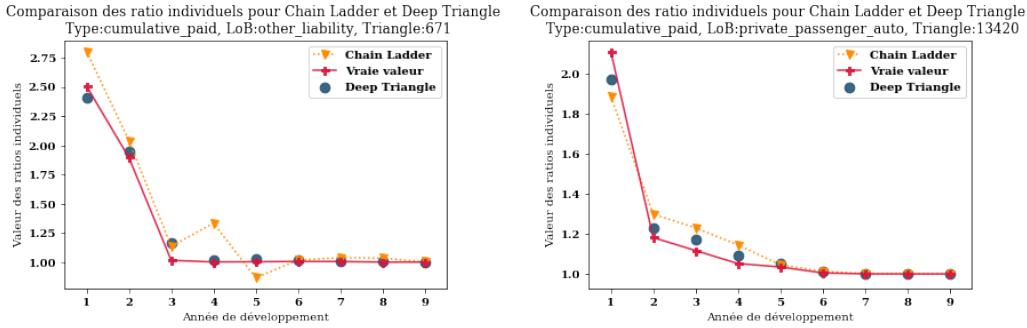
Les performances pour la prédiction à un an ont la particularité de pouvoir être observées en considérant les ratios individuels $\frac{\hat{C}_{i,j}}{C_{i,j-1}}$.

La représentation graphique de ces grandeurs nous permet d'illustrer les différences de prédiction entre le modèle *Deep Triangle* et la méthode Chain-Ladder. Nous nous sommes

2. RMSE :Root Mean Square Error.

proposée de superposer ces grandeurs sur un même graphe et de les comparer avec la valeur attendue.

FIGURE 7.1 – Comparaison des courbes représentatives de la prédition à 1 an de *Deep Triangle* et *Chain-Ladder* pour deux exemples.



La superposition des coefficients de ratios individuels nous permet une observation qualitative pour chaque entreprise de notre base de données d'étude. Ainsi, on observe que le biais est intrinsèquement maximal, quel que soit le modèle considéré, pour les premières années de développement dont les facteurs de développement sont plus volatiles. Pour l'entreprise *oth-671*, dont nous représentons les ratios individuels à gauche, on observe que *Chain-Ladder* surestime les trois premiers coefficients f_1, f_2, f_3 , puis surestime très fortement le facteur f_4 et sous-estime le facteur f_5 avant de se stabiliser. *Deep Triangle* quant à lui prédit des ratios individuels qui sont très proches de la vraie valeur bien qu'il faille noter qu'il sous-estime légèrement f_1 , et sur-estime f_3 . Pour l'entreprise *pri-13420*, les deux modèles suivent la tendance des vraies valeurs, et *Deep Triangle* se rapproche davantage des vraies valeurs. Nous verrons dans la suite que ce comportement traduit la capacité de *Deep Triangle* à lisser les estimations sur des données erratiques.

Synthèse de la performance globale pour la prédition à 1 an

Nous avons mesuré l'erreur de prédition de notre modèle *Deep Triangle* pour la prédition à 1 an. Les résultats sont les suivants :

TABLE 7.1 – Mesure de l'erreur de notre modèle pour la prédition à horizon 1 an.

	Erreur de prédition des paiements (RMSE)		Erreur de prédition des charges $\hat{paid} + \hat{os}$ (RMSE)	
	CL_Ret	DT	CL_Ret	DT
<i>com</i>	0.0652	0.0533	0.0678	0.0788
<i>oth</i>	0.0776	0.0579	0.1128	0.1217
<i>pri</i>	0.0390	0.0342	0.0272	0.0396
<i>wor</i>	0.0399	0.0375	0.1700	0.1659

Pour la prédition à horizon 1 an, les résultats de nos mesures nous indiquent que pour toutes les *LoBs*, l'erreur commise par *DT* pour la prédition des paiements est inférieure à l'erreur commise par Chain-Ladder. Pour *Deep Triangle*, la prédition de charges est obtenue en sommant les deux prédictions de paiement et de réserve. L'erreur totale qui en résulte est supérieure à l'erreur commise par Chain Ladder pour 3 *LoBs*. Les résultats de cette comparaison entre la prédition de paiements et de charges sont pertinents et peuvent être mis en perspective avec les mesures de performance à l'ultime, obtenues par application du schéma itératif de prédition, et que nous présentons dans la section suivante.

7.3 Prédiction à l'ultime

7.3.1 Mesure quantitative synthétisant la performance

7.3.1.1 Calcul de la performance globale

Par hypothèse, pour un sinistre donné, et par suite pour un agrégat de sinistres, les montants de paiements et de charges à l'ultime sont égaux. Ainsi le montant à l'ultime que l'on cherche à provisionner peut être déterminé de deux façons : on peut soit travailler à partir des données de paiements et déterminer le paiement à l'ultime, soit travailler à partir des données de charges et déterminer la charge à l'ultime.

Dans le cas de l'application de la méthode Chain-Ladder, on s'adapte au type de branche pour déterminer quelles données sont les plus appropriées à fournir un meilleur résultat. Il est fréquent de prédire à partir des triangles de charges pour les branches longues et à partir des triangles des paiements pour les branches courtes. Certains modèles, comme Munich-Chain-Ladder (QUARG & MACK, 2004), proposent l'utilisation conjointe des triangles de paiements et de charges. En outre, les résultats fournis par Chain-Ladder peuvent faire l'objet de retraitements par les actuaires, qui intègrent ainsi des jugements d'experts. Pour cette raison, il nous a semblé pertinent d'implémenter l'application de tels retraitements et d'observer également les résultats obtenus après intégration de retraitements des valeurs aberrantes. Nous détaillons dans un chapitre ultérieur (sous-sous-section 8.2.2.1) l'heuristique de retraitement que nous avons implémentée.

Dans le cas de la prédiction par notre modèle *Deep Triangle*, deux valeurs sont prédites conjointement : le montant des paiements cumulés (*paid*) à l'ultime et le montant des réserves dossier/dossier à l'ultime (*os*). En se basant sur l'hypothèse théorique d'une égalité entre les paiements cumulés à l'ultime et la charge à l'ultime, la valeur de la réserve dossier/dossier devrait être nulle ou très négligeable devant le montant des paiements cumulés. Cette hypothèse est effectivement vérifiée pour trois de nos quatre *LoBs* et n'est pas vérifiée pour la *LoB workers compensation*. Cette situation indique que pour cette *LoB*, le montant des charges dossiers/dossiers à l'ultime demeure supérieur aux prestations versées.

Nous considérons pour chacun des modèles et pour chaque *LoB* les grandeurs μ_{Paid} et $\mu_{Incurred}$, obtenues en comparant les paiements cumulés prédits et la vraie valeur des paiements d'une part, ainsi que la charge à l'ultime prédite et la vraie valeur des charges attendues d'autre part.

Les résultats obtenus sur les différentes *LoBs* sont les suivants :

TABLE 7.2 – Comparaison des prédictions de notre modèle avec la méthode de Chain-Ladder, ainsi qu’avec une approche de type Chain-Ladder incluant des retraitements des valeurs aberrantes.

	Erreurs de prédition des paiements (RMSE)			Erreurs de prédition des charges $\hat{paid} + \hat{o}s$ (RMSE)		
	<i>LoB</i>	<i>DT</i>	<i>CL</i>	<i>CL_Ret</i>	<i>DT</i>	<i>CL</i>
<i>com</i>	0,1186	0,1472	0,1438	0,1184	0,1047	0,1026
<i>oth</i>	0,1639	0,4027	0,2469	0,1784	0,2236	0,1914
<i>pri</i>	0,0518	0,0725	0,0678	0,0543	0,0503	0,0495
<i>wor</i>	0,1573	0,1743	0,1722	0,3250	0,3142	0,3190

Pour chacune des *LoBs* nous avons mis en évidence les approches conduisant aux meilleures performances.

Les résultats montrent que pour la prédition des paiements, le modèle *Deep Triangle* surpasse la méthode de Chain-Ladder au sens de l’erreur quadratique moyenne pour toutes les *LoBs*. Notamment, pour la *LoB other liability* à développement long et ayant une volatilité élevée, la RMSE du modèle de Chain-Ladder pour la prédition des paiements vaut 40% dans le cas où l’on n’inclut pas de retraitements, et 25% si l’on procède à des retraitement des coefficients, contre 16% pour le modèle *Deep Triangle*. Les résultats permettent également d’observer que pour chaque *LoB* la méthode Chain-Ladder intégrant des retraitements (*CL_Ret*) est d’une part plus performante que Chain-Ladder sans retraitements (*CL*) et d’autre part moins performante que notre modèle (*DT*), pour la prédition des paiements des entreprises de l’échantillon de test. On note aussi que sur les *LoBs com, oth* et *pri*, les performances du modèle *Deep Triangle* indiquent que l’erreur de prédition des paiements ($\hat{o}s$) et l’erreur de prédition des charges ($\hat{paid} + \hat{o}s$) sont du même ordre, ce résultat traduit une erreur très faible sur la prédition de la réserve dossier/dossier dont la valeur à l’ultime est proche de zéro.

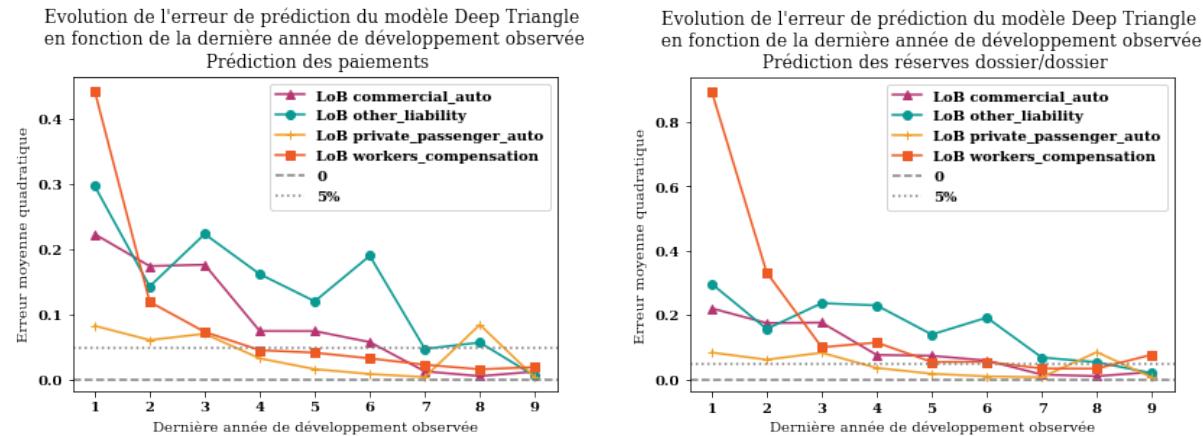
En raisonnant sur la prédition du montant à l’ultime à partir des données de paiements ou des données de charges, on observe que sur la *LoB commercial auto* ainsi que sur la *LoB private passenger auto*, à développement court, Chain-Ladder commet une erreur quadratique moyenne plus faible que le modèle *Deep Triangle*. D’autre part, on note également que dans cette situation encore, l’inclusion de retraitements permet d’obtenir de meilleurs résultats.

7.3.1.2 Impact de la durée de l’historique d’information sur la qualité des prédictions

Nous nous sommes intéressée dans un deuxième temps à l’impact que pourrait avoir la durée de l’historique observé sur la qualité de la prédition faite par le modèle *Deep*

Triangle. A cet effet nous avons réparti les agrégats de sinistres de nos échantillons de test selon le nombre d'années de développement observées, et mesuré l'erreur quadratique moyenne pour chacun de ces sous-échantillons.

FIGURE 7.2 – Représentation graphique de l'évolution de l'erreur quadratique moyenne (RMSE) de prédiction du modèle *Deep Triangle* en fonction du nombre d'années de développement observées.



L'erreur quadratique moyenne des ratios $\frac{\hat{x}_J - x_J}{NEP_i}$ est décroissante avec le nombre d'années de développement observées.

Nos mesures permettent de mettre en évidence une convergence de l'erreur moyenne quadratique vers 0. On observe la même tendance pour l'erreur moyenne.

Ainsi, les résultats de notre modèle pour la prédiction à l'ultime s'améliorent à mesure que le nombre d'années de développement observées augmente. Sur cet aspect, la méthode de Chain-Ladder présente un comportement similaire (Nous présentons en annexe (Figure B.1) les résultats obtenus pour la méthode Chain-Ladder). Cette observation est en adéquation avec l'*a priori* que l'on pourrait avoir d'une méthode de prédiction dont l'exactitude augmenterait avec la connaissance de l'historique puisque la volatilité autour du développement diminuerait.

7.3.2 Étude de la distribution des erreurs commises

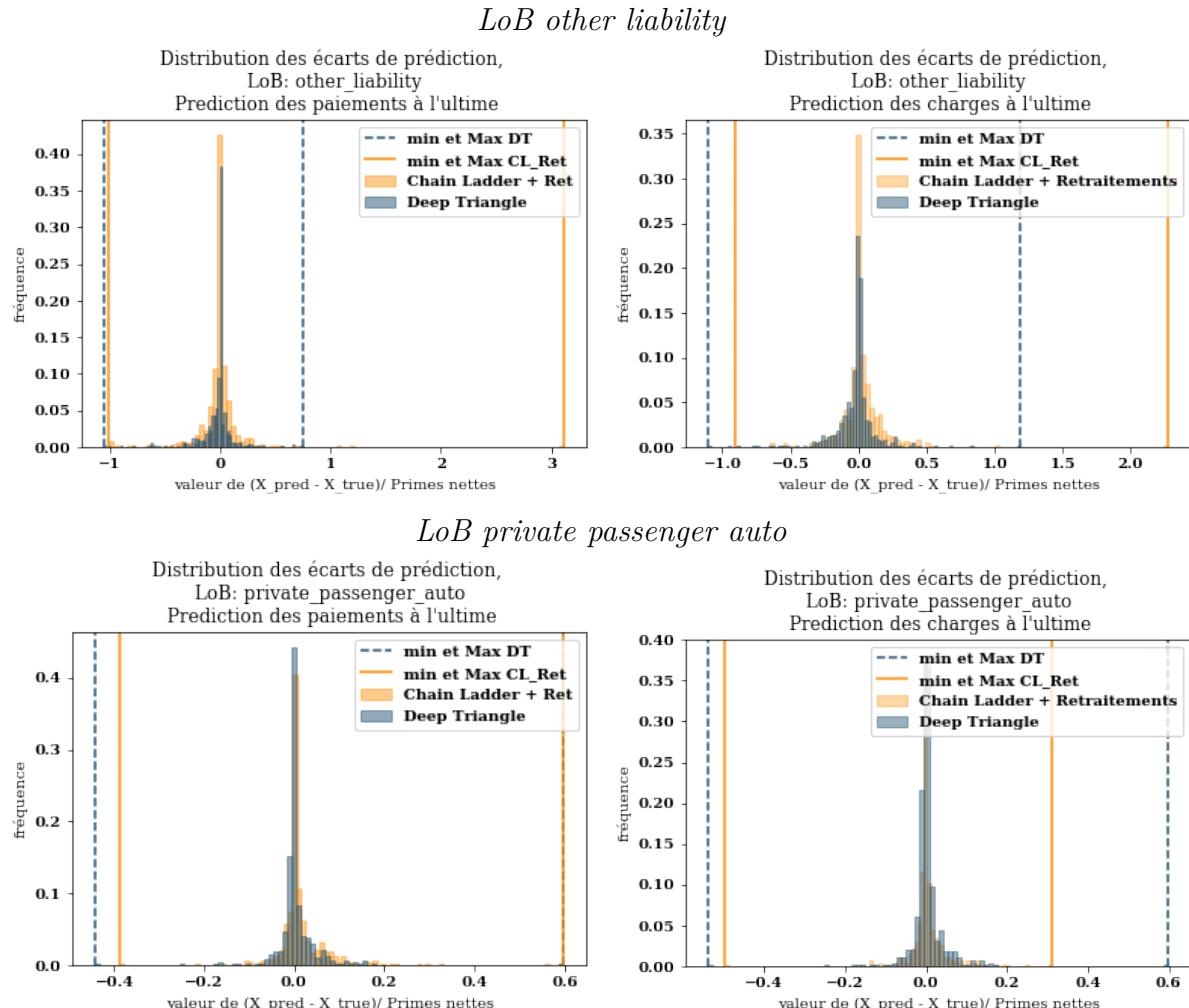
Comparaison des résultats pour deux *LoBs* très différentes

Si l'erreur moyenne quadratique est une statistique qui permet de mesurer les performances, il nous a semblé pertinent de ne pas résumer l'information et de comparer également les distributions des erreurs dans leur globalité.

Afin d'être concis, nous sélectionnons deux *LoBs* présentant des profils très différents : la *LoB other liability*, à déroulement long et volatile d'une part et la *LoB private passenger*

auto à déroulement court, plus lisse et faiblement volatile d'autre part. Les résultats pour l'ensemble des *LoBs* sont fournis en annexe (Figure B.2).

FIGURE 7.3 – Représentation de la distribution des erreurs pour deux *LoBs* très différentes.



On observe que les distributions se chevauchent et sont centrées autour de 0. Pour les prédictions des paiements de la *LoB other liability*, le support de la distribution des erreurs de *Deep Triangle* est plus restreint que celui de la distribution des erreurs de *Chain-Ladder*. Pour les prédictions des réserves de la *LoB private passenger auto*, le support de la distribution des erreurs de *Chain-Ladder* est plus restreint que celui de la distribution des erreurs de *Deep Triangle*.

Ainsi, ces résultats nous permettent de vérifier par comparaison la distribution des erreurs dans sa globalité. Les deux distributions se chevauchent, ce qui nous permet d'écartier l'hypothèse d'une inadéquation de notre modèle.

Par ailleurs cette étude met en évidence l'existence de prédictions extrêmes, pour lesquelles au moins l'un des deux modèles est très éloignée de la vraie valeur attendue.

7.3.3 Étude des prédictions dépassant un seuil de tolérance donné

En fixant un seuil de tolérance pour l'erreur δ , on peut précisément étudier tous les agrégats de sinistres pour lesquels au moins l'un des deux modèles conduit à une prédition extrême, c'est-à-dire une prédition pour laquelle l'écart entre la valeur prédite et la vraie valeur représente une part très importante du montant de primes. Le choix du seuil de tolérance approprié est propre à l'acteur qui réalise le provisionnement.

Une manière de rendre objectif le seuil peut consister à l'obtenir en déterminant pour une année donnée la distribution des ratios de pertes à l'ultime, et en retenant comme seuil la *Value at Risk* à un niveau à choisir α .

7.3.3.1 Analyse des prédictions extrêmes pour une *LoB* longue

Nous nous sommes intéressés à une analyse comparative fine des valeurs extrêmes pour la *LoB other liability*.

Détermination d'un seuil de tolérance

Afin de rendre objectif le seuil de tolérance nous avons procédé en calculant, pour une année donnée, la MSEP à l'ultime avec une formule fermée. En supposant le respect des hypothèses de Chain-Ladder par les triangles, la valeur de σ^{Mack} obtenue permet de refléter la variabilité des ratios de perte à l'ultime. Nous avons considéré un intervalle de tolérance s'étendant jusqu'à $1,5 * \sigma^{Mack}$. Ce choix correspond à un quantile de 93% sous une hypothèse paramétrique de loi Normale. Cette considération nous fournit un seuil pour nos erreurs $\delta_{C_i}^J = \frac{\hat{C}_{i,J} - C_{i,J}}{NEP_i}$.

En effet :

$$\begin{aligned} |\hat{C}_{i,J}| &\leq C_{i,J}^{\hat{Mack}} \left(1 + 1,5 \times \frac{MSEP^{Mack}}{C_{i,J}^{\hat{Mack}}} \right) \\ \iff \left| \frac{\hat{C}_{i,J} - C_{i,J}}{NEP_i} \right| &\leq \frac{1}{NEP_i} \times \left(C_{i,J}^{\hat{CL}} \left(1 + 1,5 \times \frac{MSEP^{Mack}}{C_{i,J}^{\hat{Mack}}} \right) - C_{i,J} \right) \end{aligned}$$

Nous avons mené ce travail pour les cinq entreprises les plus importantes, sélectionnées par ordre décroissant jusqu'à contribuer à plus de 80% du volume de prime émises. Nous avons pondéré les 5 seuils obtenus par l'exposition de chaque entreprise afin de déterminer notre seuil de tolérance.

Le résultat de notre étude conduit à retenir les seuils de tolérance suivants :

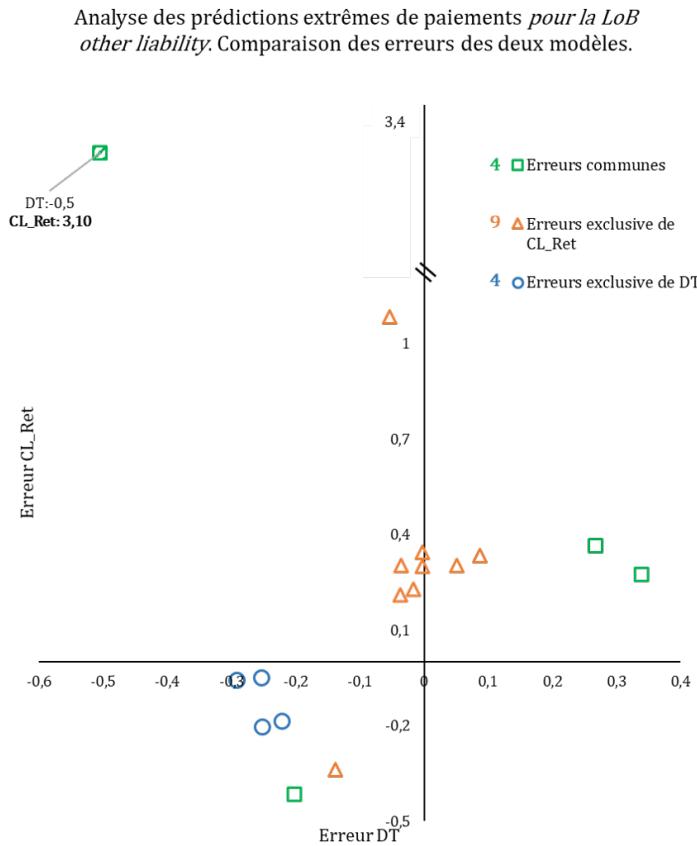
- 20% pour la prédition des paiements
- 18% pour la prédition des charges.

Comparaison des prédictions extrêmes commises par les différents modèles

Ayant déterminé les seuils de tolérance à retenir, nous pouvons mettre en évidence les performances comparées des deux modèles : d'une part notre modèle *Deep Triangle* (*DT*), et d'autre part le modèle Mack-Chain-Ladder intégrant des retraitement des valeurs aberrantes (*CL_Ret*).

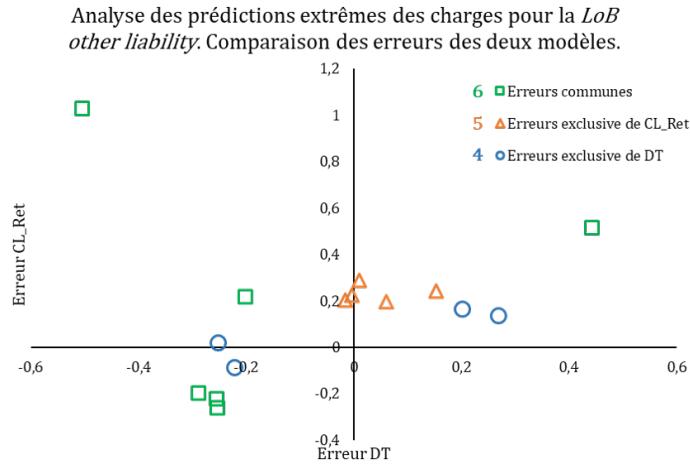
La représentation graphique que nous proposons représente tous les agrégats de sinistres ayant donné lieu à au moins une prédition extrême par l'un ou l'autre des deux modèles, dans un plan dont l'abscisse correspond à l'erreur commise par le modèle *DT*, et l'ordonnée à l'erreur commise par le modèle *CL_Ret*. Cette représentation permet de mettre en évidence les différentes situations possibles : erreur commune, erreur de *CL_Ret* et prédition acceptable de *DT* ou alors erreur de *DT* et prédition acceptable par *CL_Ret*.

FIGURE 7.4 – Représentation des prédictions extrêmes de paiements pour la *LoB other liability*.



Pour la prédiction de paiements on identifie 4 cas où les deux modèles conduisent à des prédictions extrêmes. On observe 9 cas où *CL_Ret* conduit à une prédiction extrême et pour lesquels notre modèle produit une prédiction acceptable. Enfin on observe 4 situations où notre modèle *DT* conduit à des prédictions nettement sous-évaluées (entre -20% et -30%) et où *CL_Ret* commet une erreur inférieure au seuil de tolérance , entre -20 % et 0. La nature de l'erreur est aussi importante ; ainsi, on observe notamment que bien que moins nombreuses, les erreurs exclusivement commises par *DT* correspondent à des cas de sous-provisionnement, qui affectent le caractère prudent de la provision réalisée. Pour ces agrégats de sinistres on note que la prédiction de Chain-Ladder est, elle aussi, inférieure au montant attendu.

FIGURE 7.5 – Représentation des prédictions extrêmes de charges pour la *LoB other liability*.



Pour la prédiction de charges sur la *LoB other liability* on identifie 6 cas où les deux modèles conduisent à des prédictions extrêmes. On observe 5 cas où *CL_Ret* conduit à une sur-évaluation du montant à l’ultime et pour lesquels notre modèle commet une erreur inférieure au seuil de tolérance. Enfin on observe 4 situations où notre modèle *DT* commet une erreur au-dessus du seuil de tolérance et où *CL_Ret* conduit à une prédiction acceptable.

7.3.3.2 Population des queues de distribution pour l’ensemble des *LoBs*

A titre d’exemple, nous présentons une synthèse des prédictions extrêmes au seuil de 20%.

TABLE 7.3 – Synthèses des prédictions extrêmes pour l’ensemble de *LoBs* au seuil de 20%.

Seuil de tolérance	<i>LoB</i>	Nombre total	Pourcentage de prédictions extrêmes dans notre base test	
			modèle DT	méthode CL
20%	Toutes	194	6,11%	9,78%
	<i>com</i>	39	1,28%	2,00%
	<i>oth</i>	98	3,22%	4,83%
	<i>pri</i>	12	0,17%	0,67%
	<i>wor</i>	45	1,44%	2,28%

Au seuil de 20%, 94% des prédictions de *DT* sont sous le seuil de tolérance. La méthode de Chain-Ladder conduit quant à elle à davantage de prédictions extrêmes.

Ces résultats nous permettent de conclure que quelle que soit la *LoB* concernée, le modèle *Deep Triangle* ne présente pas une tendance prononcée à fournir des prédictions

extrêmes, comparativement à la méthode de référence.

Ainsi les résultats de l'analyse des prédictions extrêmes nous permettent de conclure quant au fait que notre modèle est au moins aussi stable que le modèle de Mack-Chain-Ladder incluant des retraitements de valeurs aberrantes. Ce dernier constat, qui s'ajoute à la conclusion portant sur la mesure de performance globale, nous permet de souligner l'intérêt que pourrait revêtir le modèle *Deep triangle*, notamment pour le provisionnement des branches longues.

Ayant éprouvé les performances de notre modèle, nous nous intéressons à une analyse plus approfondie de son comportement.

Chapitre 8

Mise en évidence du comportement du réseau de neurones

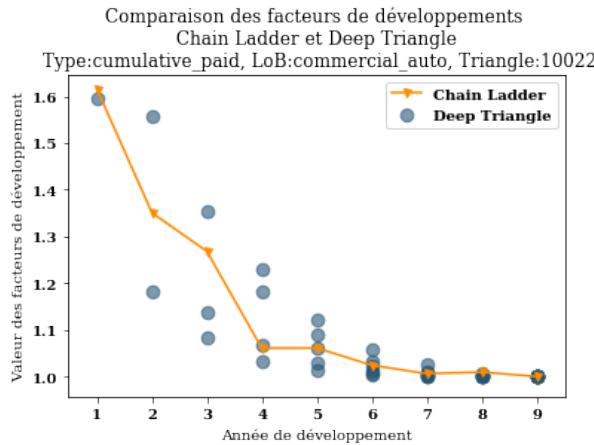
L’analyse des cas de prédiction extrême a fait ressortir des dissemblances entre notre modèle et Chain-Ladder. Nous présentons dans cette section des choix faits par notre modèle et qui diffèrent fortement des résultats proposés par Chain-Ladder. Si ces cas n’ont pas vocation de preuve, ils permettent de mettre en évidence certains comportements spécifiques à *Deep Triangle* que nous avons identifiés.

8.1 Multiplicité des facteurs de développement et articulation des informations

8.1.1 Des valeurs multiples pour les facteurs de développement en opposition à une valeur unique pour Chain-Ladder

La différence principale entre *Deep Triangle* et Chain-Ladder, réside dans la capacité du modèle *Deep triangle* à prédire plusieurs facteurs de développement pour chaque année de survenance. Le modèle de Chain-Ladder formule l’hypothèse de l’existence de facteurs de développement $f_j^{CL} = \frac{C_{i,j}}{C_{i,j-1}}$ ne dépendant que de l’année de développement et aucunement de l’année de survenance des sinistres. La méthode de Chain-Ladder conduit donc à l’obtention d’un ratio unique $\frac{C_{i,j}}{C_{i,j-1}}$ pour toutes les années de survenance i . Les facteurs $f_{j,i}^{DT} = \frac{C_{i,j}}{C_{i,j-1}}$ prédits par *Deep Triangle* dépendent à la fois de l’année de survenance et de l’année de développement. Ainsi pour chaque année de développement j , le modèle *Deep Triangle* conduit en général à l’estimation de j coefficients distincts, tandis que Chain-Ladder conduit toujours à l’estimation d’un unique coefficient.

FIGURE 8.1 – Illustration de la multiplicité des facteurs de développement de *Deep Triangle* en opposition avec l’unicité des coefficients de Chain-Ladder.

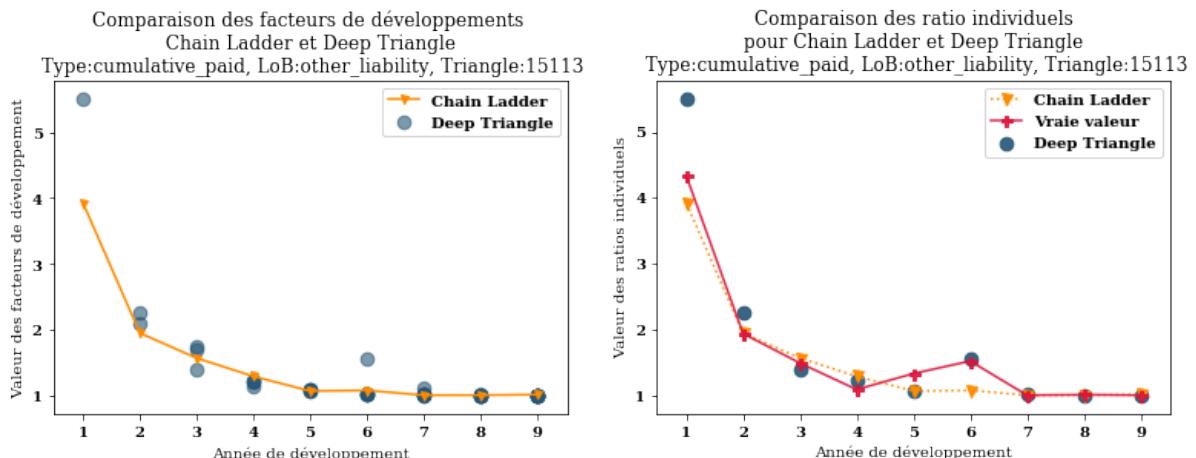


Les écarts importants entre les différents coefficients prédits par *Deep Triangle*, en font un modèle plus à même de s’adapter au développement de sinistres rattachés à différentes années de survenance, contrairement à Chain-Ladder. On l’observe notamment pour les coefficients f_2 du triangle de paiement *com-10022*.

8.1.2 Prise en compte du montant de charges

L’analyse des ratios individuels $\left(\frac{\hat{C}_{i,j}}{C_{i,j}} \right)_{j \in [i, N-i+1]}$ nous permet d’identifier une capacité de *Deep Triangle* à prendre en compte le montant de la réserve dossier/dossier afin d’ajuster sa prédition de paiements. La multiplicité des coefficients de *Deep Triangle* lui permet de tenir compte du montant de la réserve pour chaque année de survenance.

FIGURE 8.2 – Cas de prise en compte des charges par *Deep Triangle*.



Dans le cas que nous présentons, pour l'année de développement $j = 6$, la prise en compte du montant des charges permet au modèle *Deep Triangle* de produire une prédition en adéquation avec la vraie valeur observée.

Nous identifions une prédition qui se démarque des autres. Une investigation nous permet de mettre en évidence une évolution importante de la charge dans l'historique des montants. On peut ainsi conjecturer que la prise en compte des charges peut expliquer la capacité du modèle *Deep Triangle* à comprendre la nécessité d'augmenter le facteur de développement pour une année de survenance, en dépit des cinq autres valeurs qui sont plus basses.

TABLE 8.1 – Comparaison des comportements de *Deep Triangle* et de Chain-Ladder pour le triangle oth-15113 .

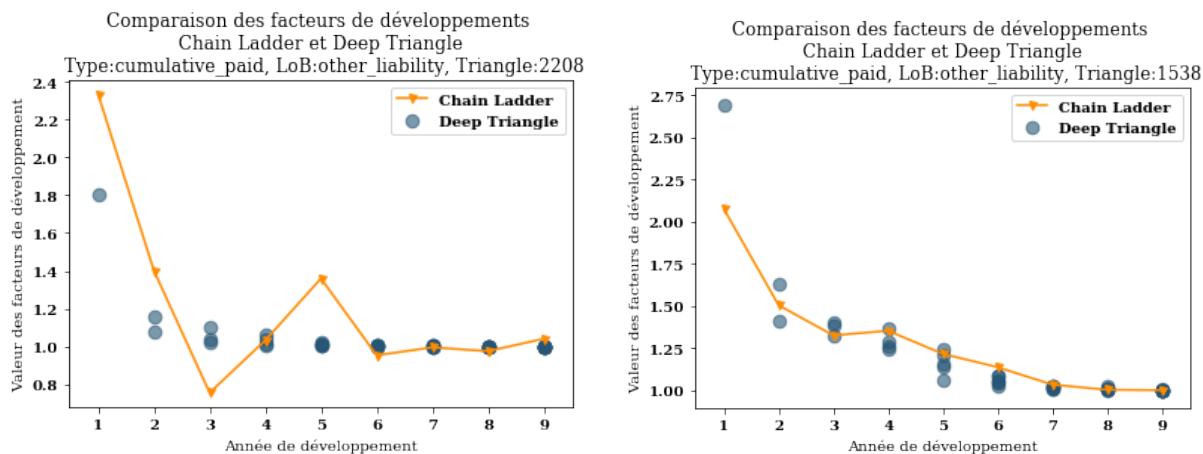
aggrégat de sinistres	Montant historiques						valeur à prédire	prédition DT	prédition CL
paid-oth-15113, 1992	33,00	57,00	85,00	177,00	182,00	189,00	287,00	293,55	202,84
incurred-oth-15113, 1992	201,00	153,00	212,00	228,00	230,00	<u>285,00</u>	308,00	329,76	285,96

Cette faculté permet au modèle *Deep Triangle* de commettre une erreur plus faible, contrairement à Chain-Ladder qui ne s'appuie que sur l'historique de paiements pour prédire les paiements futurs.

8.2 Capacité de lissage du modèle

8.2.1 Comparaison du traitement des *outliers* par *Deep Triangle* et par Chain-Ladder

FIGURE 8.3 – Mise en évidence de la capacité de lissage de *Deep Triangle*.



Confronté à ces situations, un actuaire ferait le choix de retraitier les coefficients de Chain-Ladder obtenus.

Le modèle *Deep Triangle* semble exhiber des capacités d'apprentissage lui permettant de traiter les *outliers*. Le triangle de paiements de l'entreprise oth-2208 met en évidence la capacité de lissage du modèle *Deep Triangle*. Dans ce cas, la valeur prédictive par le modèle *Deep Triangle* correspond davantage au choix de retraitement que ferait un actuaire. Pour cette raison, il convient de procéder à un retraitement des valeurs aberrantes et d'observer dans quelle mesure il permet d'améliorer les prédictions de Chain-Ladder.

8.2.2 Intégration de retraitements dans la prédiction par Chain-Ladder

8.2.2.1 Description de l'heuristique de retraitement des valeurs aberrantes

La mise en évidence de cas pour lesquels le modèle *Deep Triangle* exhibait des capacités de retraitement des *outliers* nous a conduit à affiner notre réflexion dans la comparaison du modèle *Deep Triangle* avec Chain-Ladder. A cet effet nous nous sommes proposées d'intégrer une possibilité de retraitement dans la mise en œuvre de la prédiction par Chain-Ladder. Elle articule l'utilisation du triangle de montants cumulés ($C_{i,j}$) et du triangle de ratios individuels ($R_{i,j} = \left(\frac{C_{i,j+1}}{C_{i,j}}\right)$).

Présentation de la méthode de retraitement mise en œuvre

Nous procédons de la façon suivante :

1. Nous calculons le triangle des ratios individuels à partir du triangle de montants cumulés.
2. Pour chaque année de développement j nous calculons la moyenne m_j et la variance σ_j de la distribution des ratios individuels.
3. Nous calculons les bornes $inf_j = m_j + 1,5\sigma_j$, $sup_j = m_j + 1,5\sigma_j$ et définissons l'intervalle $kept_j = [inf_j, sup_j]$.¹
4. Les ratios $R_{i_1,j}, R_{i_2,j}, \dots, R_{i_k,j}$ ne se situant pas dans l'intervalle $kept_j$ sont exclus.

$$inf_j \leq \frac{C_{i,j+1}}{C_{i,j}} \leq sup_j \iff R_{i,j} \in kept_j$$

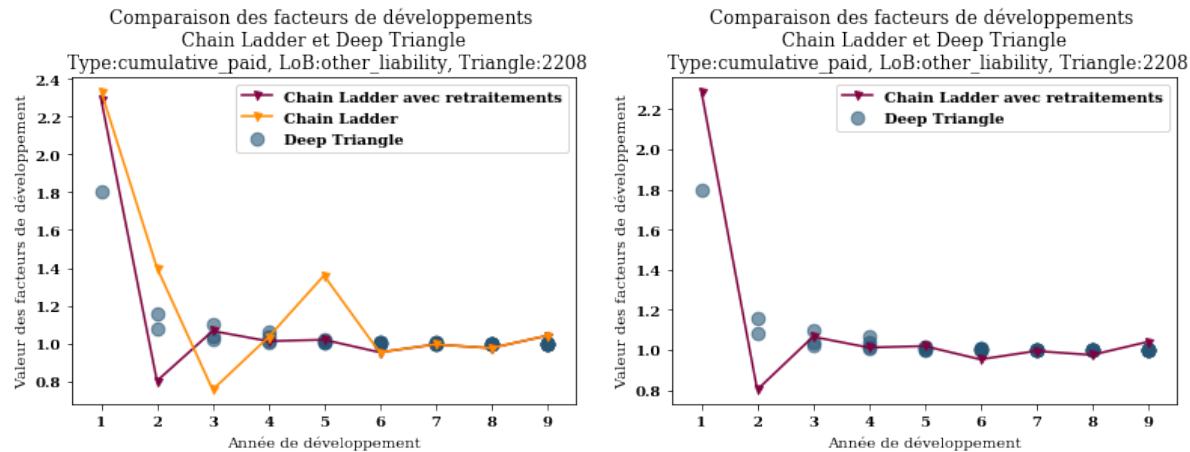
5. Dans le triangle de montant cumulés, nous calculons le coefficient \tilde{f}_j par la méthode de Chain-Ladder en excluant les montants cumulés associés aux ratios exclus.

$$\tilde{f}_j = \frac{\sum_{i=1}^{N-j} C_{i,j+1} \mathbb{1}_{R_{i,j} \in kept_j}}{\sum_{i=1}^{N-j} C_{i,j} \mathbb{1}_{R_{i,j} \in kept_j}}$$

1. Nous avons considéré la même largeur d'intervalle que dans notre étude des prédictions extrêmes.

8.2.2.2 Résultats de l'intégration de retraitements

FIGURE 8.4 – Intégration de retraitements dans la prédition de paiements du triangle oth-228 .

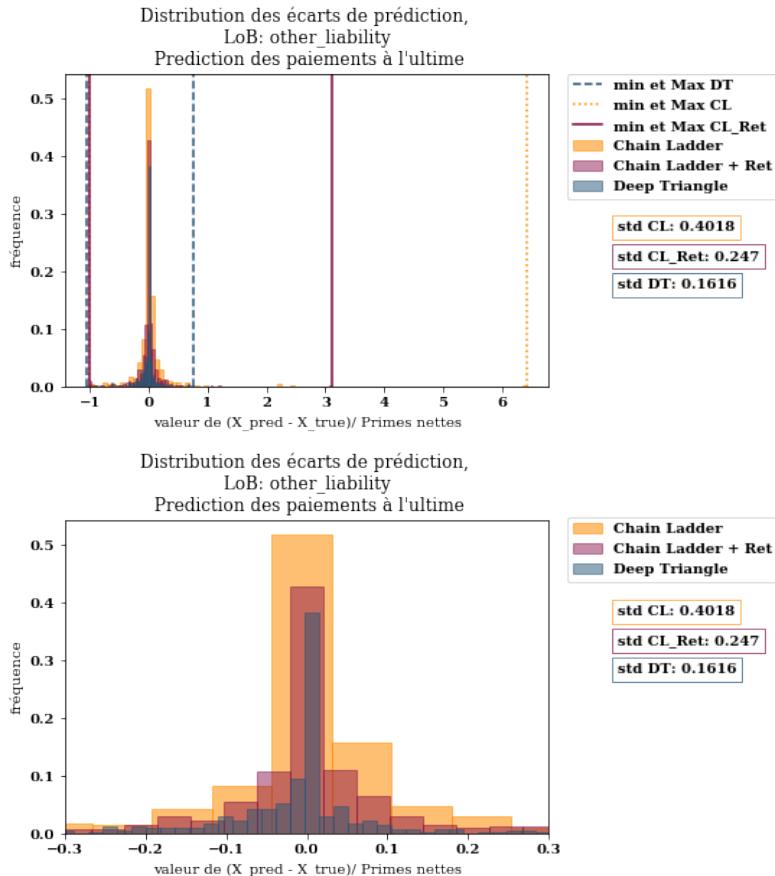


Pour le cas particulier de l'entreprise *oth-2208*, l'intégration de retraitements des valeurs aberrantes conduit à une un lissage observable pour les coefficients j_3 et j_5 .

A l'échelle globale de l'échantillon de test, l'intégration de retraitement permet d'améliorer les performances de prédition de Chain-Ladder, et ce de façon significative, comme nous l'avions souligné au cours de la mesure de performance.

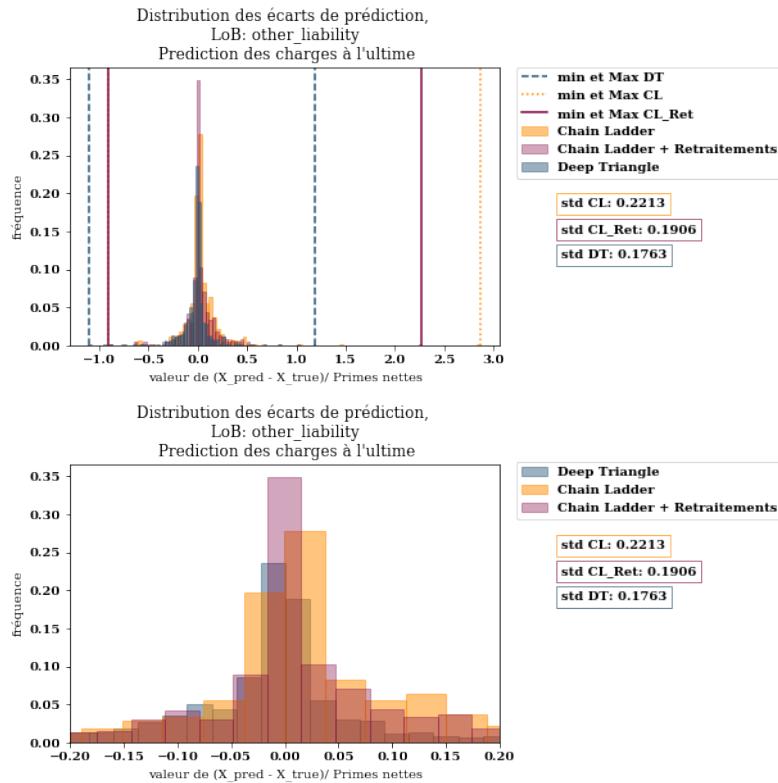
En outre l'impact sur la distribution des erreurs de prédition est observable. Afin de le mettre en évidence nous superposons les distributions d'erreurs des deux approches, conservant la distribution de notre modèle *Deep Triangle*, de façon à pouvoir jauger l'amélioration induite par le retraitement des valeurs aberrantes relativement aux performances de *Deep Triangle*. Nous présentons ici les résultats pour la *LoB other liability* les résultats pour les autres *LoBs* figurent en annexe (Figure B.2 et Figure B.3).

FIGURE 8.5 – Distributions comparées de CL_Ret et CL pour la *LoB other liability* - Prédiction des paiements.



Pour la prédiction de paiements, on observe que l'inclusion de retraitements dans la méthode de Chain-Ladder nous permet d'aboutir à une distribution d'erreurs de prédiction dont le support est plus restreint. L'écart-type de la distribution s'en trouve diminué, passant 0.40 à 0.25. Notamment, l'intégration des retraitements dans la prédiction permet de se prémunir d'une prédiction extrême pour lequel le ratio d'erreur de Chain-Ladder sans retraitement valait 6. Les deux approches sont cependant surpassées par le modèle *Deep Triangle*.

FIGURE 8.6 – Distributions comparées de CL_Ret et CL pour la *LoB other liability* - Prédiction des charges.



Pour la prédiction des charges à l'ultime, l'inclusion de retraitements permet également de restreindre le support de la distribution des erreurs de prédiction et de contourner 1 prédiction extrême. L'impact sur l'écart type de la distribution est dans ce cas plus nuancé que pour la prédiction des paiements. L'erreur commise par *Deep Triangle* est quant à elle plus faible.

Ainsi, bien que l'intégration de retraitements dans la prédiction par Chain-Ladder permette d'améliorer les résultats, la capacité de traitement des valeurs aberrantes que semble exhiber notre modèle *Deep Triangle* lui permet tout de même de surpasser les performances de Chain-Ladder sur notre *LoB* longue *other liability*.

Chapitre 9

Étude et interprétation du comportement du modèle

Ayant mis en évidence des comportements spécifiques exhibés par le modèle nous nous proposons de l'étudier davantage afin d'en tirer une meilleure compréhension. A cet effet nous étudions différents aspects du comportement prédictif du modèle.

9.1 Étude de la sensibilité du modèle à la couche *embedding*

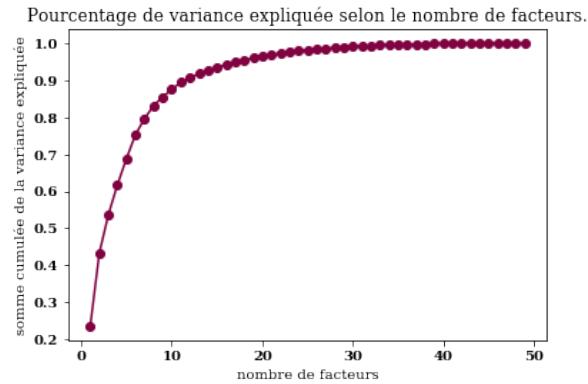
9.1.1 Analyse de la spatialisation réalisée par la couche *embedding*

Afin d'approfondir notre compréhension des résultats obtenus, nous nous sommes proposée d'étudier dans un premier temps la spatialisation de nos entreprises réalisée par la couche *embedding*. Pour cette étude nous avons sélectionné la *LoB other liability*, qui est à déroulement long.

9.1.1.1 Analyse en composantes principales des vecteurs obtenus

Nous avons pris l'initiative dans un premier temps de donner du sens aux vecteurs de l'espace *embedding* en réalisant une analyse en composantes principales des vecteurs. Dans le développement de notre étude, nous avons été confortée dans l'intérêt de la démarche par des d'autres études statistiques de couches *embedding* à l'instar de celles menées par BOLUKBASI et al. (2016) ou par MUSIL (2019).

FIGURE 9.1 – Somme du pourcentage de variance expliquée en fonction du nombre d'axes factoriels.

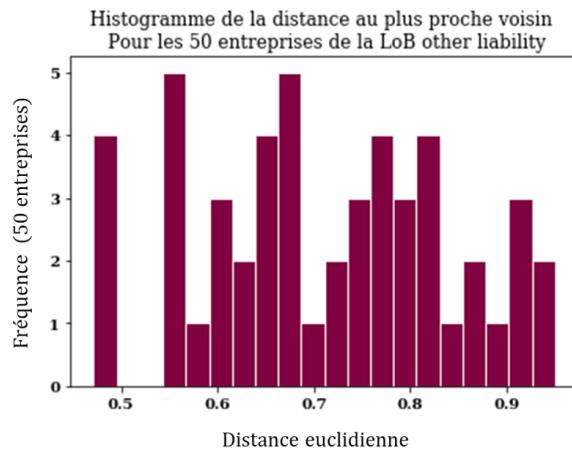


L'analyse en composantes principales que nous avons menée n'a pas été concluante car la proportion de variance expliquée par les premiers axes vaut 45% : cette quantité est trop faible pour nous permettre de conférer un pouvoir explicatif à ces axes.

9.1.1.2 Exploration de la proximité entre chacun des groupes d'assurance pour une *LoB*

Dans un deuxième temps nous nous sommes proposées d'analyser les proximités euclidiennes de nos vecteurs en vue d'observer des similitudes dans le déroulement des sinistres ou les caractéristiques des entreprises qu'ils représentent. A cet effet nous avons construit une matrice de distance à partir des vecteurs issus de leur projection spatiale dans l'espace d'*embedding*.

FIGURE 9.2 – Histogramme de la distance au plus proche voisin dans l'espace d'*embedding*.



Nous avons ainsi pu identifier les deux entreprises les plus proches pour la *LoB other liability*. Nous avons, par exemple, pu vérifier l'absence de lien immédiat motivant la

proximité mais avons été confrontée à une limite liée à la dimension des données et au nombre important de paramètres possiblement significatifs. Ainsi les similitudes ayant conduit à un rapprochement par la couche *embedding* n'ont pas pu être révélées par l'ACP que nous avons menée ni par notre deuxième approche.

9.1.2 Étude de la sensibilité du modèle à l'usage de la couche *embedding*

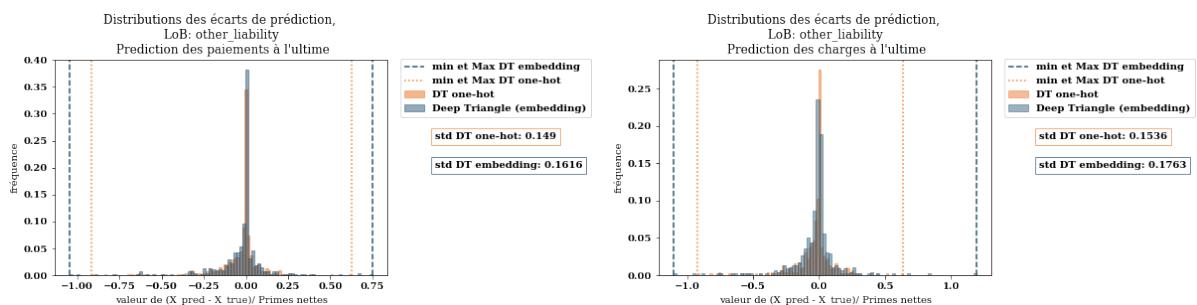
Nous nous sommes proposée d'étudier par la suite la sensibilité du modèle à la couche *embedding*, en supprimant celle-ci et en la remplaçant par un encodage *one-hot*¹. Par cette modification, nous obtenons ainsi un réseau de neurones présentant une architecture différente, que nous entraînons avec les mêmes données de sinistres pour réaliser la même tâche de prédiction que *Deep Triangle*. La statistique résumant la performance indique donc des performances relativement comparables, voire un peu meilleures pour le modèle *DT one-hot*.

TABLE 9.1 – Comparaison des performances de prédiction de *DT one-hot* et *Deep Triangle* à la couche *embedding*.

modèle	Prédiction des paiements	Prédiction des charges
<i>DT one-hot</i>	0.1532	0.1571
<i>Deep Triangle (embedding)</i>	0.1639	0.1784

Nous nous sommes également intéressée à la distribution des erreurs, les résultats obtenus sont les suivants :

FIGURE 9.3 – Distribution des erreurs de prédiction du modèle issu de la substitution d'un encodage *one-hot* à la couche *embedding*.



Pour la *LoB other liability*, le réseau de neurones *DT one-hot* produit des résultats similaires à ceux du réseau de neurones *Deep Triangle (embedding)*. Il est toutefois à

1. L'encodage *one hot* est une représentation vectorielle booléenne des modalités d'une grandeur prenant un nombre fini de valeurs distinctes.

noter une distribution d'erreurs plus resserrée pour le réseau de neurones *DT one-hot*, en particulier pour la prédiction des charges à l'ultime. Plus précisément, on observe 3 prédictions extrêmes commises par *Deep Triangle (embedding)*, qui dépassent toutes les valeurs prédites par *DT one-hot*. Si l'encodage *one-hot* ne détériore pas les performances de notre modèle pour la *LoB other liability*, il est important de souligner que dans le cas de notre étude nous disposons de données d'entraînement suffisantes pour chaque entreprise. Il serait pertinent de pouvoir mesurer si dans le cas d'une entreprise aux données disponibles moindres, le rapprochement avec d'autres entreprises, réalisé uniquement par l'usage de la couche *embedding*, pourrait être utile pour une meilleure prédiction de montant futurs.

Cette étude nous permet donc de conclure que pour une *LoB* telle qu'*other liability*, la substitution d'un encodage *one-hot* à la couche *embedding* dans l'architecture du modèle, peut contribuer à contourner ou atténuer certaines prédictions extrêmes, et résulter en une amélioration de la statistique de performance globale.

Ayant consacrée une étude de sensibilité à la couche *embedding* qui prend en argument l'identifiant de l'entreprise, nous nous intéressons dans la suite aux résultats dépendant de la donnée historique de montants fournie en entrée.

9.2 Étude de l'impact de la dynamique de déroulement du sinistre dans la prédiction

En vue d'explorer le comportement du modèle quant à la dynamique de déroulement, nous avons fait l'expérience de perturber les valeurs d'un triangle de ratios de paiements incrémentaux de façon à conserver inchangée la valeur des paiements cumulés.

Déroulement de l'étude

A cette fin, nous avons considéré trois scénarios de perturbation à partir d'une valeur ϵ . Nous avons ensuite réalisé des prédictions à partir des nouveaux historiques de montants incrémentaux ainsi obtenus. Nous présentons les résultats pour deux valeurs de ϵ distinctes. Nous rappelons également les prédictions obtenues en l'absence de perturbation.

TABLE 9.2 – $\epsilon = 0.05$: mise en évidence de la prise en compte de la dynamique du déroulement.

Historique	incrément $j - 3$	incrément $j - 2$	incrément $j - 1$	prédiction de l'incrément de paiement en j	prédiction du total de réserve en j
fidèle	0.1630	0.2205	0.1107	0.0475	0.1244
perturbé(1)	$0.1630 + \epsilon$	$0.2205 - \epsilon$	0.1107	0.0484	0.1268
perturbé(2)	$0.1630 - \epsilon$	0.2205	$0.1107 + \epsilon$	0.0453	0.1223
perturbé(3)	0.1630	$0.2205 + \epsilon$	$0.1107 - \epsilon$	0.0491	0.1238

TABLE 9.3 – $\epsilon = 0.2$: mise en évidence de la prise en compte de la dynamique du déroulement.

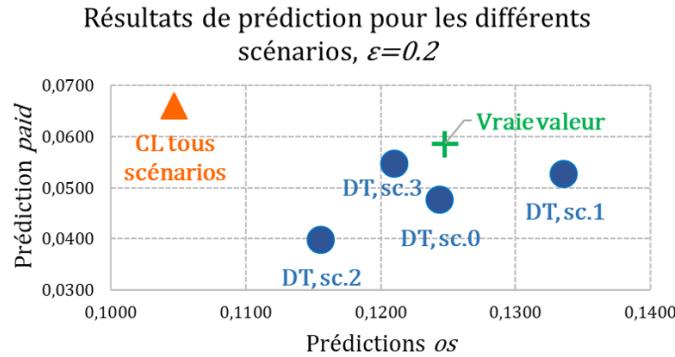
Historique	incrément $j - 3$	incrément $j - 2$	incrément $j - 1$	prédition de l'incrément de paiement en j	prédition du total de réserve en j
fidèle	0.1630	0.2205	0.1107	0.0475	0.1244
perturbé(1)	0.1630 + ϵ	0.2205 - ϵ	0.1107	0.0526	0.1336
perturbé(2)	0.1630 - ϵ	0.2205	0.1107 + ϵ	0.0397	0.1156
perturbé(3)	0.1630	0.2205 + ϵ	0.1107 - ϵ	0.0545	0.1211

Analyse des résultats

Chaque scénario conduit à une prédition différente des autres.

Les résultats distincts que nous obtenons pour chaque perturbation nous permettent de mettre en évidence la prise en compte de la dynamique du sinistre par le modèle *Deep Triangle*. Ces résultats illustrent une différence importante par rapport à la méthode de Chain-Ladder qui s'applique sur les paiements cumulés et dans le cas de laquelle les trois scénarios conduisent à une valeur identique.

FIGURE 9.4 – Comparaison des prédictions de *Deep Triangle* et Chain-Ladder pour chaque scénario.



On souffre toutefois d'un manque de compréhension des raisons pour lesquelles les prédictions évoluent dans chaque scénario.

9.3 Sensibilité à l'introduction de perturbations dans les historiques de sinistres

Afin de pouvoir mieux appréhender le comportement de notre modèle, nous avons tenté d'étudier l'impact de chacune des données d'entrée sur la prédiction.

A cet effet nous nous sommes proposés de perturber marginalement chacune des composantes du vecteur des montants historiques et d'en mesurer l'impact sur la prédiction.

Comment choisir les données à perturber ?

Nous avons perturbé de véritables historiques présents dans notre base de données de test. Nous avons choisi un agrégat de sinistres pour lequel les résultats du modèle *Deep Triangle* étaient cohérents² afin de pouvoir nous fier à l'évolution des résultats induite par la perturbation des montants historiques. Pour les sinistres regroupés dans l'agrégat que nous avons choisi, la dernière année de développement observée est l'année 3. La *LoB* concernée est *workers' compensation*.

TABLE 9.4 – Aperçu des composantes du vecteur X_1 avant perturbation, la valeur 99. est la valeur de *masking* du réseau de neurones.

historique de montants	<i>paid ratios</i>	<i>os ratios</i>
j-1	0.11070459	0.19874897
j-2	0.22058303	0.28750148
j-3	0.16298832	0.52283725
j-4	99.	99.
j-5	99.	99.
j-6	99.	99.
j-7	99.	99.
j-8	99.	99.
j-9	99.	99.

Nous notons P_j la valeur des paiements incrémentaux au cours de l'année de développement $j \in [1, 4]$, et OS_j le montant de réserve dossier/dossier pour l'année de développement $j \in [1, 4]$.

Déroulement de l'étude

On cherche à prédire le vecteur $\hat{Y} = [\hat{P}_j, \hat{OS}_j]$ à partir du code identifiant l'entreprise et des données $X_1 = [P_{j-3}, P_{j-2}, P_{j-1}, OS_{j-3}, OS_{j-2}, OS_{j-1}]$.

Nous nous proposons d'observer l'influence marginale de chacune des composantes du vecteur X_1 sur la prédiction \hat{Y} . Pour cela nous faisons varier chaque composante sur

2. Nous nous sommes restreints à des erreurs de prédiction inférieures à 3% du volume de primes, ce pourcentage est arbitraire mais cohérent.

l'intervalle $[-0.5, 0.5]$ en substituant un bruit ϵ à sa valeur. Nous perturbons individuellement les données historiques de prédiction en remplaçant une des composantes du vecteur X_1 par ϵ . Pour chacune de ces 6 perturbations, nous visualisons la courbe représentative de \hat{P}_j et \hat{OS}_j en fonction de la valeur de ϵ .

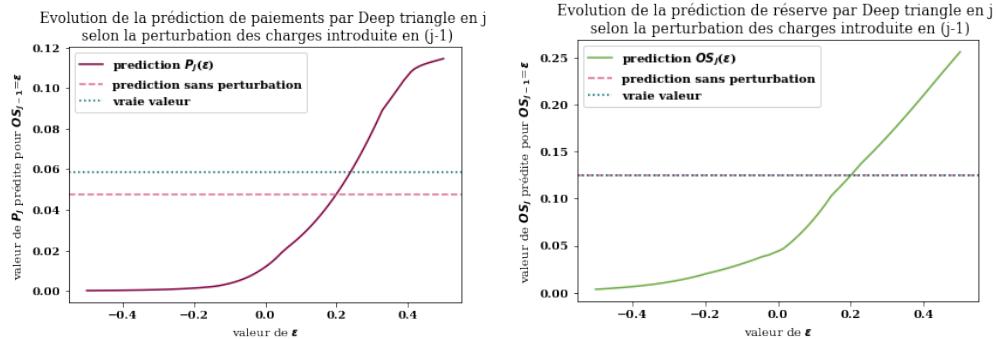
Analyse du résultat

Nous fournissons en annexe les résultats obtenus pour chacune des 6 perturbations (section C.1).

Les courbes représentatives de l'impact marginal de chacun de ces 6 paramètres mettent en lumière plusieurs éléments.

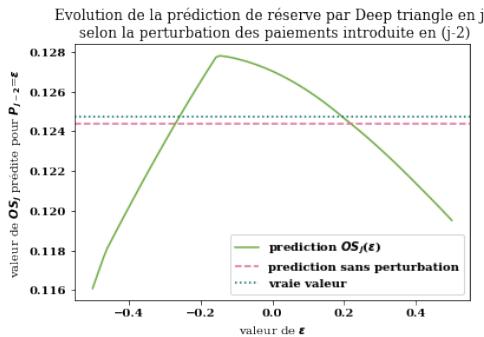
Tout d'abord une perturbation d'un montant historique de paiements ou de réserve produit des effets à la fois sur la prédiction du montant de paiements incrémentaux et sur la prédiction du montant de réserve. Ce résultat confirme l'utilisation conjointe par *Deep triangle* des deux historiques pour la prédiction des montants futurs.

FIGURE 9.5 – Courbes représentatives de l'impact marginal d'une perturbation des montants de réserve sur la prédiction de chaque composante de Y .



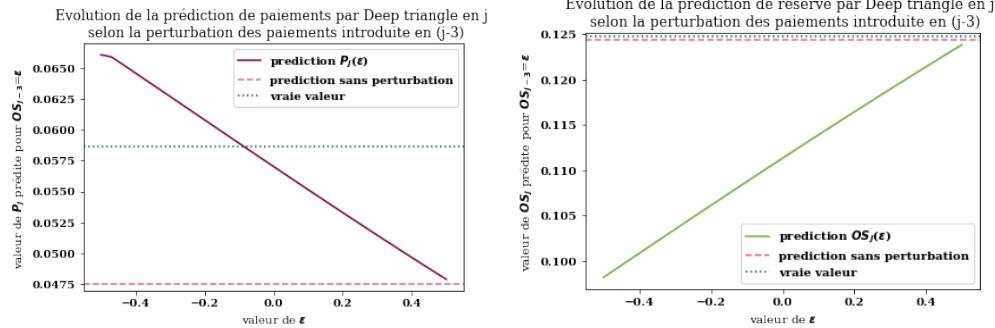
Un deuxième point mis en évidence par notre étude est que l'impact de ces coefficients n'est pas nécessairement monotone.

FIGURE 9.6 – Mise en évidence des cas de non-monotonie de l'évolution marginal de Y .



Enfin on observe que la perturbation des montants relatifs à l'année $j - 3$ semble avoir un impact quasi-linéaire sur les prédictions pour l'année j dans le cas de notre exemple.

FIGURE 9.7 – Mise en évidence de l'impact de la variation des réserves à $j - 3$ sur la prédiction en j .



L'observation résultant de notre étude nous permet de mesurer et caractériser l'évolution des prédictions Y selon la valeur des composantes de X_1 .

Ainsi l'analyse menée nous permet de mettre en évidence différents aspects de la prise en compte de l'historique dans la prédiction du modèle. Elle ne nous permet toutefois pas une compréhension des choix du modèle. A cet effet, nous nous proposons l'utilisation de l'outil *LIME* d'aide à l'interprétabilité.

9.4 Utilisation de *LIME*, outil d'aide à la compréhension en *Machine Learning*

Afin d'être à même d'interpréter les choix faits par le modèle nous avons décidé de réaliser une prédition en utilisant *LIME*³ un *framework* Python d'aide à l'interprétabilité présenté par RIBEIRO et al. (2016a).

9.4.1 Fonctionnement de *LIME*

LIME offre un cadre qui accompagne les prédictions. Il nécessite :

- un régresseur : ici la prédition de paiements ou de réserve.
- un ensemble de poids : ici le réseau de neurones entraîné.
- une instance à partir de laquelle réaliser la prédition.

Afin d'acquérir une capacité d'interprétabilité pour une prédition, *LIME* approxime de façon locale le modèle de prédition général. A cet effet, le cadre *LIME* induit des perturbations des données d'entrée et en mesure l'impact. Le résultat obtenu permet de mettre exergue les variables qui contribuent significativement à la prédition, ainsi que les modalités de cette contribution.

9.4.2 Résultats d'explication d'une prédition

Nous avons souhaité réaliser une prédition pour un agrégat de sinistre de la *LoB other liability*, à développement long.

Les sorties de *LIME* se lisent comme suit :

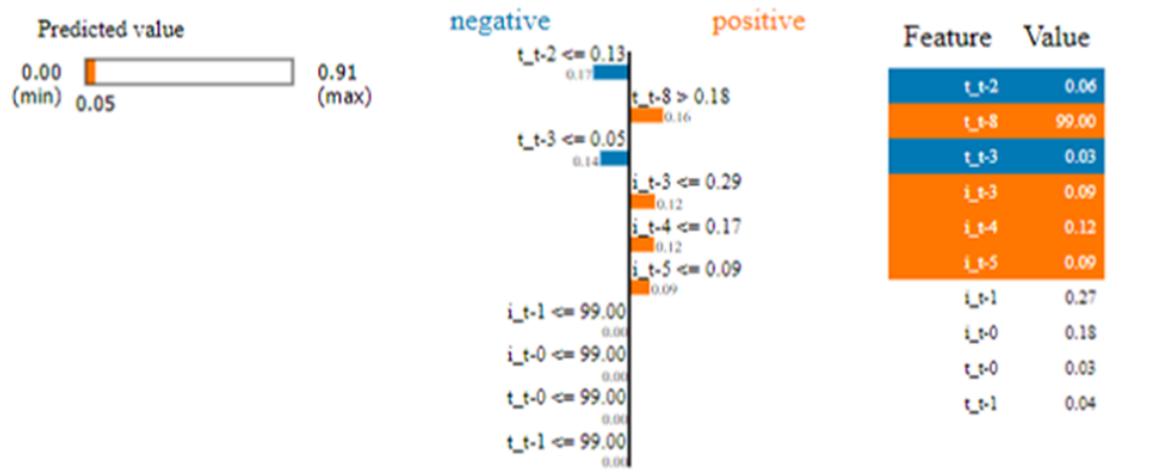
notation <i>LIME</i>	variable correspondante
$t_t - j$	$\mapsto paid_{t-(9-j)}$
$i_t - j$	$\mapsto os_{t-(9-j)}$

Cette ré-indexation tient à l'ordre chronologique des données et est nécessaire pour bien comprendre les schémas.

3. L.I.M.E : Local Interpretable Model-agnostic Explanations.

Le caractère *model agnostic* de *LIME* tient à ce qu'il a été implémenté en vue de pouvoir être employé pour tout type de classifieur ou de régresseur.

FIGURE 9.8 – Sortie produite par *LIME* pour la prédition de paiement incrémental de oth-14370



Les sorties obtenues avec *LIME* nous indiquent l’effet marginal de chacun des montants historiques sur la sortie prédite par le modèle. Nous avons choisi d’observer les résultats des 10 variables impactant le plus la prédition. Le schéma nous indique que 6 variables en particulier ont un impact non neutre sur la prédition.

Pour la prédition réalisée avec *LIME* que nous présentons en Figure 9.8, les résultats obtenus nous indiquent que :

- $paid_{t-7} \leq 0.13$ a un impact négatif sur la prédition. La valeur de masquage étant par convention 99., la condition $paid_{t-7} \leq 0.13$ indique que le sinistre était déjà survenu il y a 7 ans et que le paiement incrémental était inférieur à 13% du volume de primes net. Compte tenu du fait que le réseau de neurones a été entraîné sur des sinistres que l’on suppose développés en 10 ans ou moins, cette condition apparaît cohérente.
- $paid_{t-1} > 0.18$ a impact positif sur la prédition. Cette situation correspond à une valeur très importante pour l’année précédente, ou éventuellement inconnue (99).
- $paid_{t-6} \leq 0.05$ a un impact négatif sur la prédition.
- $ost_{t-6} \leq 0.29$, $ost_{t-5} \leq 0.17$ et $ost_{t-4} \leq 0.09$ ont un impact positif sur la prédition.

Ces résultats nous permettent de formuler une interprétation possible de ces conditions selon laquelle on comprend qu’une connaissance de la relative ancienneté du sinistre (déjà survenu il y a 7 ans ou il y a 6 ans) conduit à diminuer la prédition de paiement incrémental en t , tandis qu’une absence de valeur pour l’année précédente conduit à une augmentation de la prédition du montant de paiement incrémental. *LIME* nous fournit des seuils quantitatifs précis à partir desquels l’impact des variables explicatives sur la prédition est modifié. Ces valeurs nous permettent de comprendre localement une prédition réalisée par notre réseau de neurones compte tenu de l’historique qui lui a été présenté et des valeurs des poids du réseaux de neurones, résultat de la phase d’entraînement. Par ailleurs

l'utilisation de *LIME* met aussi en évidence que le montant de charges impacte de façon significative la prédiction de paiement.

Ainsi l'outil *LIME* nous permet de comprendre des choix de notre modèle de provisionnement pour une prédiction choisie. On peut identifier quelles variables ont eu un impact significatif dans la prédiction et quantifier quels seuils ont conduit à évaluer à la hausse ou la baisse le montant prédit.

Cependant, en dépit des multiples éléments de compréhension qu'apporte ce cadre de prédiction, les résultats fournis par *LIME* ne sont pas pleinement satisfaisants. Notamment la valeur des seuils est propre à l'agrégat de sinistres pour lequel on cherche à prédire les montants à l'ultime. Nous sommes en mesure de proposer une potentielle explication des sorties de *LIME*, un travail reste à faire afin d'être en mesure de formaliser une mise en relation entre des caractéristiques assurantielles du déroulement des agrégats de sinistres et les conditions précises portant sur les variables explicatives, obtenues par l'utilisation de *LIME*. Afin d'avancer dans ce sens il serait pertinent de pouvoir recourir à un *cadre* qui synthétiseraît de façon exhaustive les résultats de *LIME* pour l'ensemble des agrégats de sinistres de notre base de données, en vue d'acquérir la capacité d'exprimer, par exemple, ces seuils non pas de façon absolue mais relativement aux montants du déroulé des sinistres.

Conclusion

Notre étude nous a permis d'implémenter un modèle de réseau de neurones récurrent à même d'être employé pour prédire conjointement les montants futurs de paiements et de réserves de dossier/dossier à un horizon temporel de 10 ans.

Nous avons montré que ce modèle est à même de rivaliser avec une approche de type Mack-Chain-Ladder. Les résultats du modèle nous montrent la précision supplémentaire qu'il apporte dans la prédiction de montants pour les branches longues et volatiles comparé à une méthode de référence telle que Chain-Ladder. De plus, l'étude de la distribution des erreurs des deux modèles nous permet de confirmer que les résultats de prédiction de notre modèle *Deep Triangle* sur les *LoBs* courtes et très peu volatiles sont du même ordre que les résultats obtenus avec la méthode Chain-Ladder appliquée à des triangles de paiements ou de charges. Enfin, l'étude des queues de distribution que nous avons menée, nous permet de mettre en évidence que le modèle *Deep Triangle* est au moins aussi stable que la méthode Chain-Ladder qui conduit à davantage de prédictions extrêmes, y compris lorsque l'on intègre à l'approche Mack-Chain-Ladder des retraitements des valeurs aberrantes. Ainsi, en considérant l'unique critère de la qualité des résultats qu'il propose, il apparaît que l'emploi du modèle *Deep Triangle* pour la prédiction conjointe des paiements et des charges pour des branches d'activité IARD est envisageable.

La compréhension des prédictions du modèle est un autre critère que notre travail a examiné. Dans cet objectif, une étude approfondie du fonctionnement du modèle nous a permis dans un premier temps de mettre en évidence les particularités de notre modèle en comparaison de Chain-Ladder. Ainsi, il ressort que la prédiction d'un montant de charges ou de paiements est affectée par les historiques des deux quantités. Plus encore, notre modèle est caractérisé par une prise en compte de la dynamique du déroulement des sinistres, là où la méthode de Chain-Ladder, qui ne considère que les montants cumulés, n'est pas en mesure de déceler et d'exploiter ces informations. De plus, conduisant à la prédiction de multiples facteurs de développement pour une même année, *Deep Triangle* montre une capacité à articuler les informations de charges et de paiements dans sa prédiction de montants futurs. Enfin, il apparaît que les capacités d'apprentissage du modèle lui permettent de traiter les *outliers*. Dans ces cas, une comparaison nous indique que les performances de Mack-Chain-Ladder incluant des retraitements sont surpassées par *Deep Triangle*, notamment sur une *LoB* longue. Le modèle exhibe ainsi des comportements de lissage, qui sont intéressants car ils conduisent dans certains cas à des résultats qui ne nécessitent pas de retraitement.

La mise en évidence des spécificités du fonctionnement du modèle *Deep Triangle*, nous a conduit à chercher des moyens d'améliorer nos capacités d'interprétabilité des choix du

modèle : nous avons donc eu recours à un outil d'aide à l'interprétabilité afin de caractériser plus quantitativement les sensibilités de notre modèle aux variables explicatives. Cet accompagnement de nos prédictions par l'outil *LIME* nous a permis d'acquérir la capacité de déterminer localement les variables explicatives impactant significativement la prédiction, ainsi que les modalités de leur influence sur la prédiction. Nous sommes également à même d'obtenir une quantification des valeurs seuils pour ces variables explicatives, conduisant à une augmentation ou une baisse du montant prédit par notre modèle. Cet outil nous apporte ainsi des éléments de compréhension pour chaque prédiction de notre modèle. Toutefois les seuils quantitatifs et l'ordre d'importance des variables sont propres à chaque prédiction. Notre capacité de compréhension est ainsi limitée par la difficulté, à ce stade, de rattacher toutes les conditions portant sur les variables explicatives à une interprétation assurantielle, et de traduire à une échelle macro la capacité de compréhension locale acquise. Pour avancer dans ce sens, une piste pertinente serait de pouvoir acquérir une capacité d'explicabilité globale du modèle, en exprimant par exemple les seuils en fonction des montants figurant dans l'historique. En l'absence d'un pouvoir explicatif plus global, il peut sembler trop tôt pour envisager d'employer notre modèle comme modèle de provisionnement, compte tenu des exigences du régulateur quant à l'interprétabilité des modèles.

Au terme de notre étude, il apparaît qu'un usage de notre modèle est envisageable à des fins d'automatisation du provisionnement. Notre modèle pourrait par exemple être utile dans le cadre de processus de revue indépendante de provisionnement. On pourrait nourrir une base de données dans le but d'avoir un outil permettant de challenger un modèle de provisionnement, dont les résultats pourraient être employés afin de mettre en évidence les *LoBs* nécessitant une revue détaillée. Une autre perspective d'emploi du modèle réside dans l'élargissement de la modélisation des sinistres graves, pour lesquels on dispose effectivement d'historiques de charges et de développements, et dont on cherche à améliorer l'estimation de la charge ultime. On pourrait envisager d'adapter le stock de données que nous avons employé, avec des données de sinistres graves dont la maille d'agrégation pourrait être plus fine, possiblement ligne à ligne. Enfin, compte tenu du caractère spécifique des données employées pour calibrer notre modèle, il serait pertinent de pouvoir mettre à l'épreuve l'architecture *Deep Triangle* avec des données d'assureurs français, et de considérer une étude se concentrant sur des *LoBs* à risque telles que la responsabilité civile médicale ou la responsabilité civile construction.

Références bibliographiques

Livres

- LECUN, Y. Et al. (1998). Efficient BackProp. In *Neural Networks : Tricks of the Trade* (p. 9-48). Springer-Verlag.
- MINSKY, M. & PAPERT, S. (1969). *Perceptrons : An Introduction to Computational Geometry.*

Articles

- BENGIO, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *CoRR, abs/1206.5533*.
- BOLUKBASI, T. Et al. (2016). Man is to Computer Programmer as Woman is to Home-maker ? Debiasing Word Embeddings. *CoRR, abs/1607.06520*.
- BORNHUECKER, R. L. & FERGUSON, R. E. (1972). The actuary and IBNR. *Proceedings of the Casualty Actuarial Society, 59*, 181-195.
- BRAUN, H. & LAI, L. L. (2005). A Neural Network Linking Process for Insurance Claims. *International Conference on Machine Learning and Cybernetics*.
- BROCKETT, P. L. Et al. (1997). A Case Study in Applying Neural Networks to Predicting Insolvency for Property and Casualty Insurers. *The Journal of the Operational Research Society, 48*, 1153-1162.
- CHO, K. Et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Conference on Empirical Methods in Natural Language Processing*.
- GABRIELLI, A. (2019). A Neural Network Boosted Double Over-Dispersed Poisson Claims Reserving Model. *ssrn/3365517*.
- GABRIELLI, A. & WÜTHRICH, M. V. (2018). An Individual Claims History Simulation Machine. *Risk, 6/2*, 29.
- HAREJ, B. Et al. (2017). Individual Claim Development with Machine Learning. *ASTIN working party*, 1-36.
- HOCHREITER, S. & SCHMIDHUBER, J. (1997). Long Short-Term Memory. *Neural Comput., 9*(8), 1735-1780.
- HORNIK, K., STINCHCOMBE, M. & WHITE, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359-366.

- JAMAL, S. Et al. (2018). Machine Learning and Traditional Methods Synergy in Non-Life Reserving. *ASTIN working party*, 38-41.
- KINGMA, D. P. & LEI BA, J. (2015). Adam : a méthod for stochastic optimization. *International Conference on Learning Representations*.
- KUO, K. (2018). Deep Triangle : A Deep Learning Approach to Loss Reserving. *arXiv*, 1804.09253.
- MACK, T. (1993). Distribution-free calculation of the standard error of chain Ladder reserve estimates. *ASTIN Bulletin*, 23(2), 213-226.
- MULQUINEY, P. (2006). Artificial Neural Networks in Insurance Loss Reserving. *9th Joint International Conference on Information Sciences (JCIS-06)*.
- MUSIL, T. (2019). Examining Structure of Word Embeddings with PCA. *CoRR*, abs/1906.00114.
- QUARG, G. & MACK, T. (2004). Munich Chain Ladder : A Reserving Method that Reduces the Gap between IBNR Projections Based on Paid Losses and IBNR Projections Based on Incurred Losses. *Blätter der Deutschen Gesellschaft für Versicherungs- und Finanzmathematik*, 26, 597-630.
- REDDI, S. J., KALE, S. & KUMAR, S. (2018). On the Convergence of Adam and Beyond. Proceedings of 2018 International Conference on Learning Representations. *CoRR*, abs/1904.09237.
- RIBEIRO, M. Et al. (2016a). "Why Should I Trust You ?" : Explaining the Predictions of Any Classifier. *CoRR*, abs/1602.04938.
- ROSENBLATT, F. (1958). The perceptron : a probabilistic model for information storage and organizationin the brain. *Psychological Review*, 65, 6.
- RUMELHART, D. E., HINTON, G. E. & WILLIAMS, R. J. (1986). Learning Representations by Back-propagating Errors. *Parallel distributed processing : explorations in the microstructure of cognition*, 1, 318-382.
- WÜTHRICH, M. V. (2018). Neural Networks Applied to Chain-Ladder Reserving. *European Actuarial Journal*, 2, 407-436.
- CHOLLET, F. Et al. (2015). Keras.
- RIBEIRO, M. Et al. (2016b). LIME.

Table des figures

Sauf mention d'une source externe, les figures ont été réalisées par l'auteure du mémoire.

1	Architecture du réseau <i>Deep Triangle</i>	iv
2	Deep Triangle network architecture	ix
1.1	Évolution chronologique de la charge et des paiements d'un sinistre.	8
3.1	Courbes représentatives de trois fonctions d'activation pouvant être utilisées.	22
3.2	Représentation d'un réseau de neurones dense <i>feedforward</i> à une seule couche cachée.	28
3.3	Aperçu d'un réseau de neurones récurrent.	29
3.4	Représentation d'un <i>RNN</i> déroulé.	30
3.5	Schéma d'un <i>GRU</i>	31
3.6	Schéma de la mise en cascade de plusieurs unités <i>GRU</i>	33
4.1	Architecture du réseau de neurones <i>DeepTriangle</i>	40
5.1	Représentation de la multiplicité des activités des groupes d'assurance de notre base de données d'étude.	45
5.2	Comparaison de l'évolution des coefficients de Mack-Chain-Ladder de chacun des secteurs d'activité : triangles de paiements.	46
5.3	Comparaison de l'évolution des coefficients de Mack-Chain-Ladder de chacun des secteurs d'activité : triangles de charges.	47
5.4	Comparaison des caractéristiques de deux <i>LoBs</i> : triangles de paiements.	48
5.5	Comparaison des caractéristiques de deux <i>LoBs</i> : triangles de charges.	49
5.6	Analyse des ratios de perte des entreprises de notre base de données d'étude.	50
5.7	Représentation graphique de la part des primes des assureurs les plus importants, au seuil de 80% de part de primes cumulées.	51
6.1	Illustration de la mise à date des données.	53
6.2	Utilisation des données de sinistres pour la réalisation de l'entraînement.	54
6.3	Présentation du fonctionnement du schéma itératif implémenté.	58
7.1	Comparaison des courbes représentatives de la prédition à 1 an de <i>Deep Triangle</i> et Chain-Ladder pour deux exemples.	63
7.2	Représentation graphique de l'évolution de l'erreur quadratique moyenne (RMSE) de prédition du modèle <i>Deep Triangle</i> en fonction du nombre d'années de développement observées.	67
7.3	Représentation de la distribution des erreurs pour deux <i>LoBs</i> très différentes.	68
7.4	Représentation des prédictions extrêmes de paiements pour la <i>LoB other liability</i>	71

7.5	Représentation des prédictions extrêmes de charges pour la <i>LoB other liability</i> .	72
8.1	Illustration de la multiplicité des facteurs de développement de <i>Deep Triangle</i> en opposition avec l'unicité des coefficients de Chain-Ladder.	76
8.2	Cas de prise en compte des charges par <i>Deep Triangle</i> .	76
8.3	Mise en évidence de la capacité de lissage de <i>Deep Triangle</i> .	77
8.4	Intégration de retraitements dans la prédiction de paiements du triangle oth-228	79
8.5	Distributions comparées de CL_Ret et CL pour la <i>LoB other liability</i> - Prédiction des paiements.	80
8.6	Distributions comparées de CL_Ret et CL pour la <i>LoB other liability</i> - Prédiction des charges.	81
9.1	Somme du pourcentage de variance expliquée en fonction du nombre d'axes factoriels.	84
9.2	Histogramme de la distance au plus proche voisin dans l'espace d' <i>embedding</i> .	84
9.3	Distribution des erreurs de prédiction du modèle issu de la substitution d'un encodage <i>one-hot</i> à la couche <i>embedding</i> .	85
9.4	Comparaison des prédictions de <i>Deep Triangle</i> et Chain-Ladder pour chaque scénario.	87
9.5	Courbes représentatives de l'impact marginal d'une perturbation des montants de réserve sur la prédiction de chaque composante de Y .	89
9.6	Mise en évidence des cas de non-monotonie de l'évolution marginal de Y .	89
9.7	Mise en évidence de l'impact de la variation des réserves à $j - 3$ sur la prédiction en j .	90
9.8	Sortie produite par <i>LIME</i> pour la prédiction de paiement incrémental de oth-14370	92
A.1	Analyse des ratios de perte des entreprises de notre base de données d'étude	103
A.2	Comparaison des caractéristiques de nos différentes <i>LoBs</i> : triangles des paiements.	104
A.3	Comparaison des caractéristiques de nos différentes <i>LoBs</i> : triangles de charges	105
B.1	Représentation graphique de l'évolution de l'erreur <i>RMSE</i> de prédiction du modèle Chain-Ladder en fonction du nombre d'années de développement observées.	107
B.2	Effets de l'inclusion des retraitements au modèle de Mack-Chin-Ladder : prédiction des paiements	108
B.3	Effets de l'inclusion des retraitements au modèle de Mack-Chin-Ladder : prédiction des charges	109

Liste des tableaux

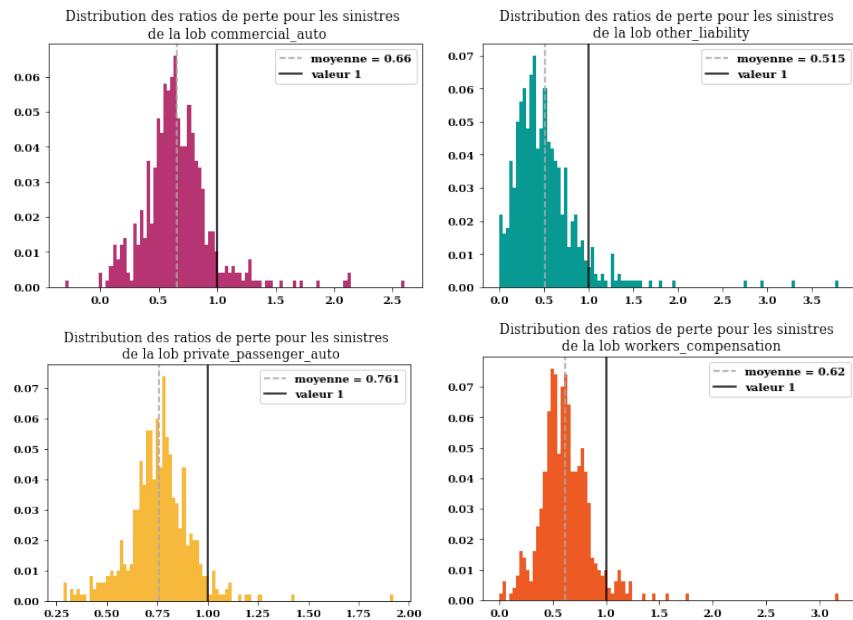
1	Comparaison des prédictions de notre modèle avec la méthode de Chain-Ladder ainsi qu'avec une approche de type Chain-Ladder incluant des retraitements des valeurs aberrantes.	v
2	Comparison of prediction performance including our model, the Chain-Ladder method and a Chain-Ladder approach performing outlier reprocessing.	ix
2.1	Exemple de triangle de liquidation. I=4, J=4	14
2.2	Application de la méthode Chain-Ladder à un exemple.	16
2.3	Prévisions obtenues à l'ultime par la méthode Chain-Ladder.	17
3.1	Présentation de trois exemples de fonctions d'activation.	22
3.2	Exemples de fonctions de perte pour la régression et expression de la fonction de coût associée.	24
3.3	Comparaison d'un RNN standard et d'un <i>GRU</i> pour la mise à jour de l'état caché <i>hidden state</i>	32
5.1	Présentation des variables décrivant une entrée de notre base de données. .	44
7.1	Mesure de l'erreur de notre modèle pour la prédiction à horizon 1 an. . . .	64
7.2	Comparaison des prédictions de notre modèle avec la méthode de Chain-Ladder, ainsi qu'avec une approche de type Chain-Ladder incluant des retraitements des valeurs aberrantes.	66
7.3	Synthèses des prédictions extrêmes pour l'ensemble de <i>LoBs</i> au seuil de 20%. <td>72</td>	72
8.1	Comparaison des comportements de <i>Deep Triangle</i> et de Chain-Ladder pour le triangle oth-15113	77
9.1	Comparaison des performances de prédiction de <i>DT one-hot</i> et <i>Deep Triangle</i> à la couche <i>embedding</i>	85
9.2	$\epsilon = 0.05$: mise en évidence de la prise en compte de la dynamique du déroulement.	86
9.3	$\epsilon = 0.2$: mise en évidence de la prise en compte de la dynamique du déroulement.	87
9.4	Aperçu des composantes du vecteur X_1 avant perturbation, la valeur 99. est la valeur de <i>masking</i> du réseau de neurones.	88

Annexe A

Complément de statistiques descriptives

A.1 Analyse des ratios de perte en distinguant les différents secteurs d'activité

FIGURE A.1 – Analyse des ratios de perte des entreprises de notre base de données d'étude



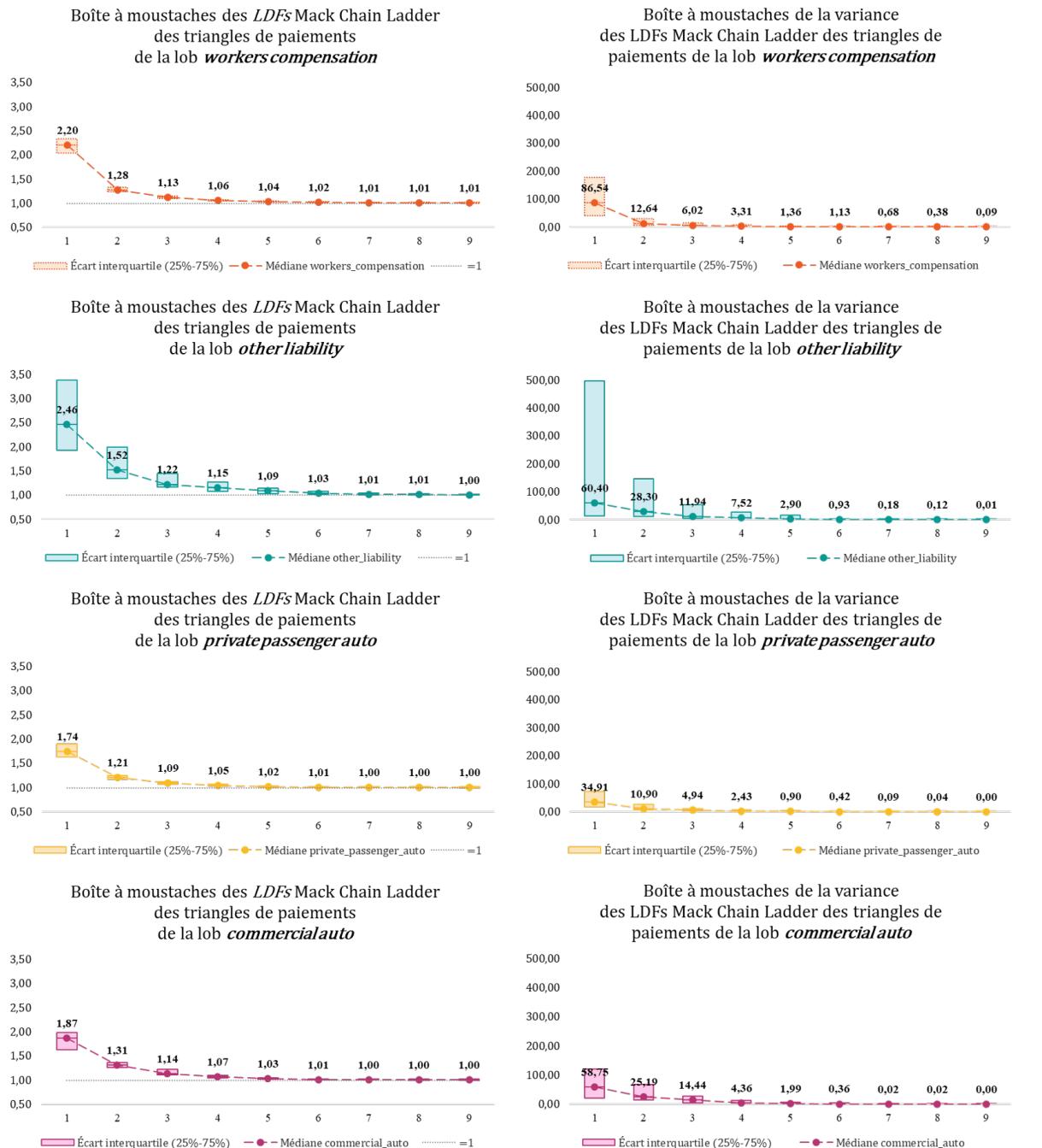
Nous représentons séparément la distribution des ratios de perte des sinistres pour chacune des LoBs.

On observe que les ratios de perte sont sensiblement plus faibles pour la *LoB other liability*.

Dans cette situation, la substitution de P à la valeur inconnue de S à l'ultime dans le calcul des ratios d'erreur nous conduit à multiplier l'erreur par un facteur <2. Cette mesure reste donc acceptable d'autant qu'elle s'applique de façon similaire aux deux modèles que l'on cherche à comparer.

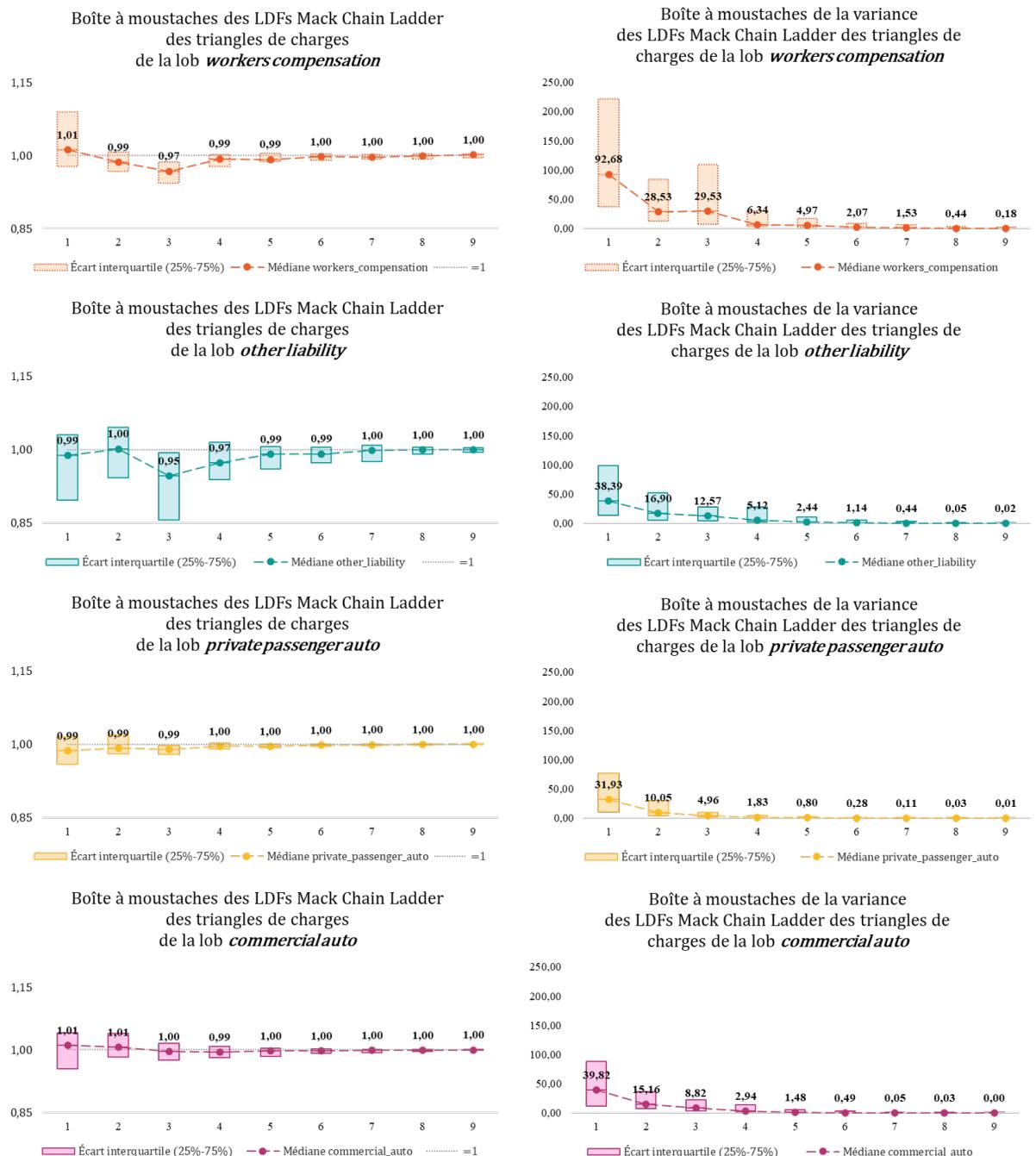
A.2 Analyse des facteurs de développement médians et leur variance sur chaque *LoB*

FIGURE A.2 – Comparaison des caractéristiques de nos différentes *LoBs* : triangles des paiements.



La comparaison de ces valeurs nous permet de mettre en évidence la typologie des différentes *LoBs*.

FIGURE A.3 – Comparaison des caractéristiques de nos différentes *LoBs* : triangles de charges

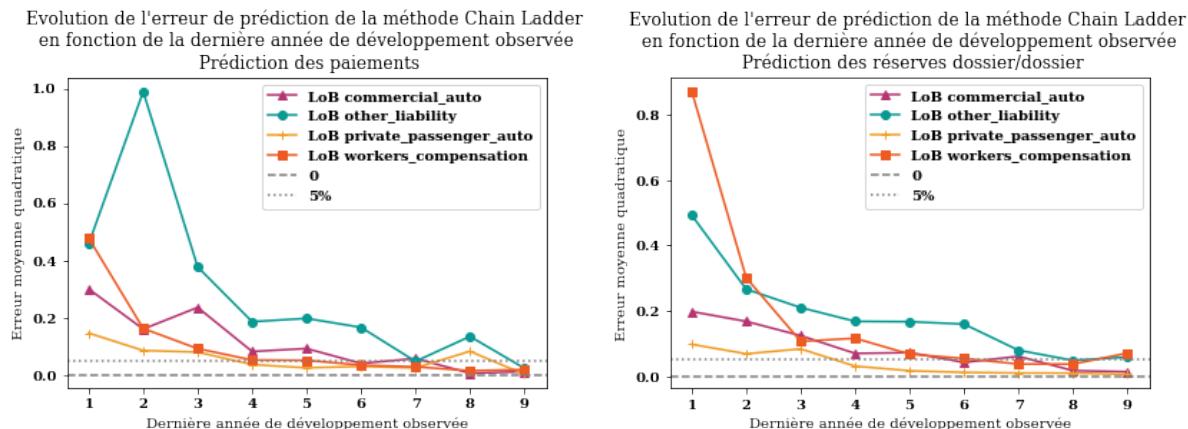


Annexe B

Annexes aux résultats de mesure des performances

B.1 Mise en évidence de l'impact du *lag* initial sur l'erreur

FIGURE B.1 – Représentation graphique de l'évolution de l'erreur *RMSE* de prédiction du modèle Chain-Ladder en fonction du nombre d'années de développement observées.



Comme expliqué dans notre analyse des résultats du modèle *Deep Triangle*, on observe aussi pour Chain-Ladder une amélioration des prédictions en fonction du nombre d'années de développement observées.

B.2 Distributions empiriques des erreurs de prédiction et résultats de l'inclusion de retraitements dans la prédiction par Mack-Chain-Ladder

FIGURE B.2 – Effets de l'inclusion des retraitements au modèle de Mack-Chain-Ladder : prédiction des paiements

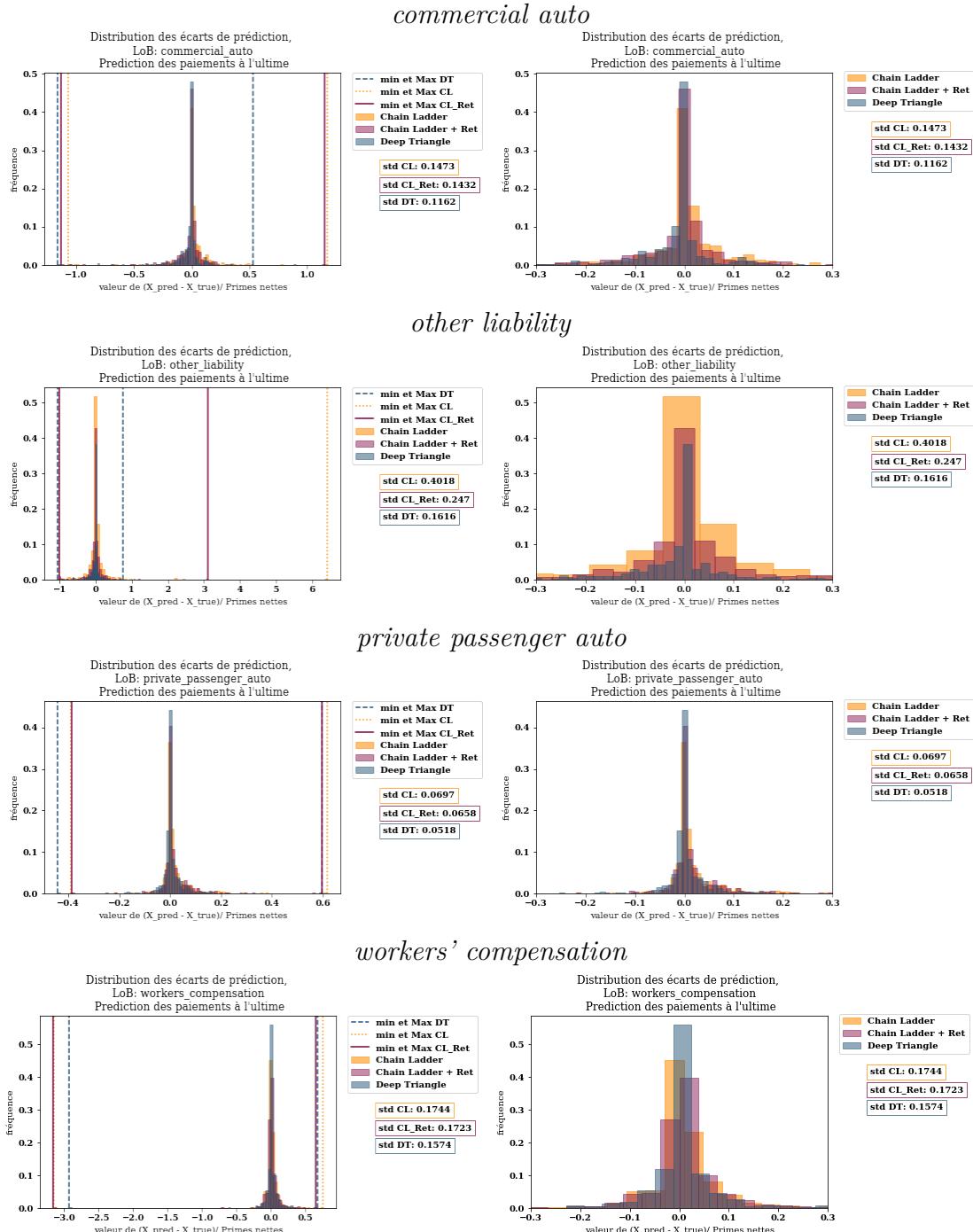
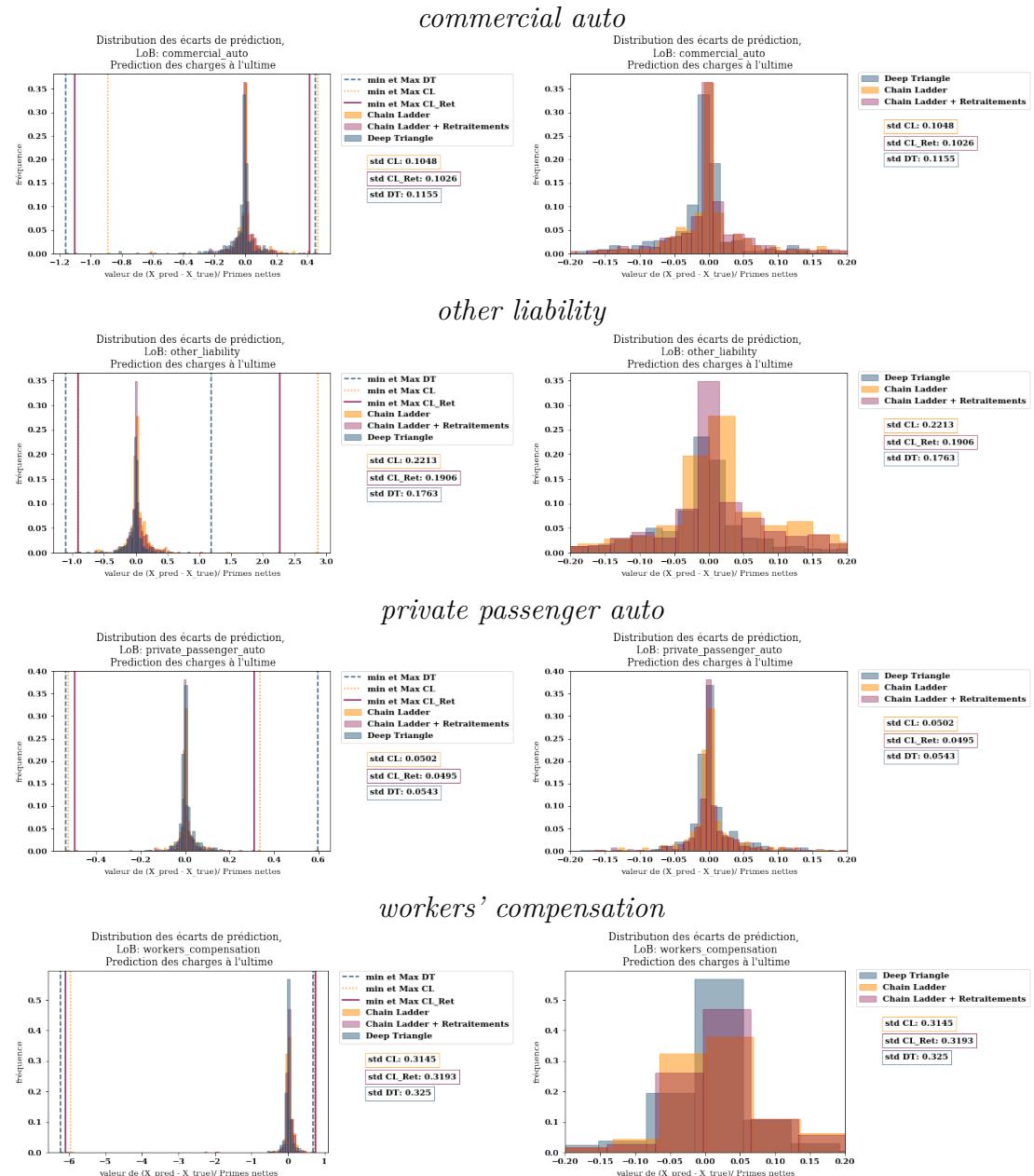


FIGURE B.3 – Effets de l'inclusion des retraitements au modèle de Mack-Chin-Ladder : prédiction des charges



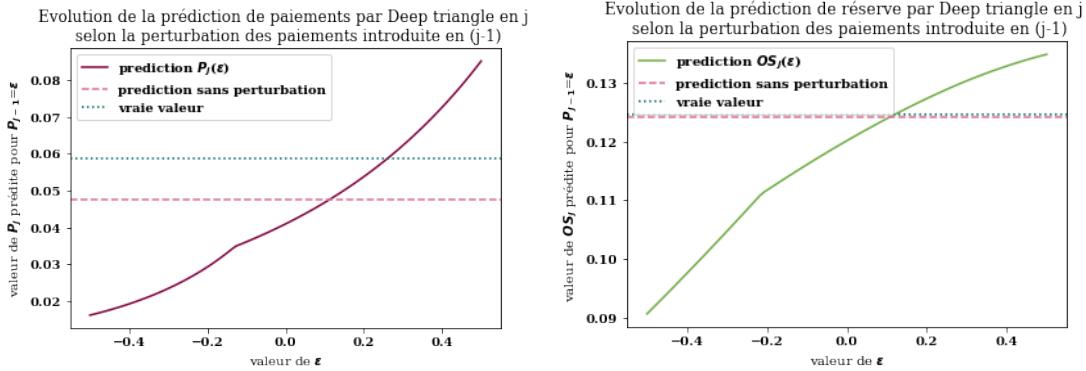
Annexe C

Mise en évidence du comportement du modèle *Deep Triangle*

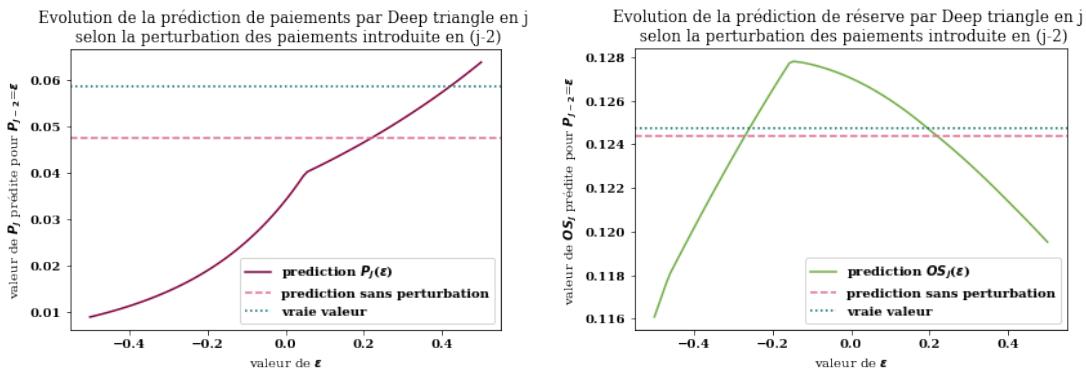
C.1 Courbes représentatives de l'impact marginal des données historiques sur la prédiction

Représentation graphique de $\hat{P}_j(\epsilon)$ et $\hat{OS}_j(\epsilon)$ pour $\epsilon \in P_{j-3}, P_{j-2}, P_{j-1}, OS_{j-3}, OS_{j-2}, OS_{j-1}$

Effets d'une perturbation du montant des paiements en $j - 1$

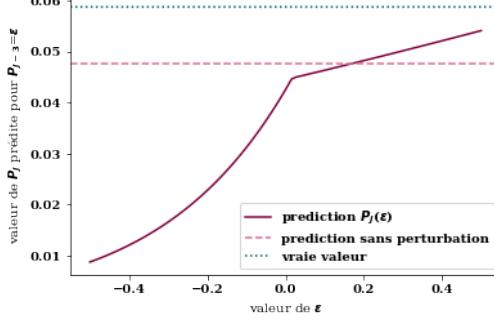


Effets d'une perturbation du montant des paiements en $j - 2$

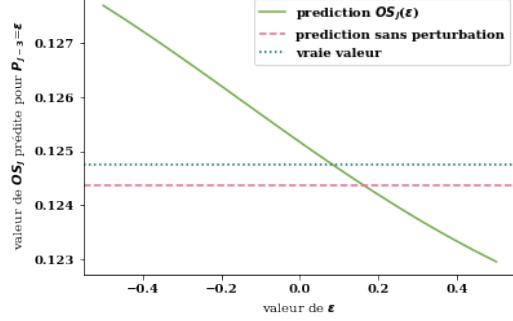


Effets d'une perturbation du montant des paiements en $j - 3$

Evolution de la prédiction de paiements par Deep triangle en j selon la perturbation des paiements introduite en ($j - 3$)

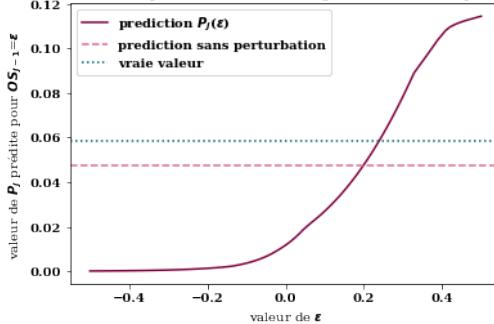


Evolution de la prédiction de réserve par Deep triangle en j selon la perturbation des paiements introduite en ($j - 3$)

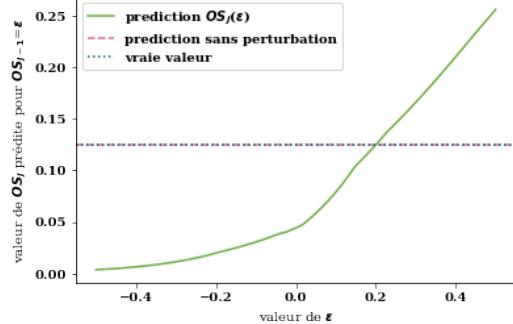


Effets d'une perturbation du montant des réserves en $j - 1$

Evolution de la prédiction de paiements par Deep triangle en j selon la perturbation des charges introduite en ($j - 1$)

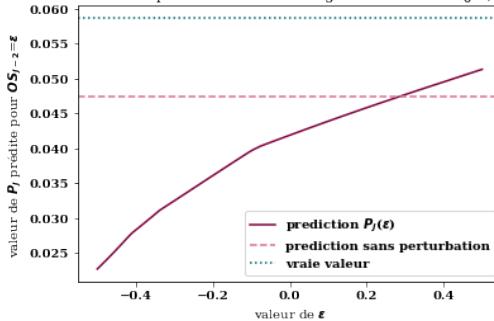


Evolution de la prédiction de réserve par Deep triangle en j selon la perturbation des charges introduite en ($j - 1$)

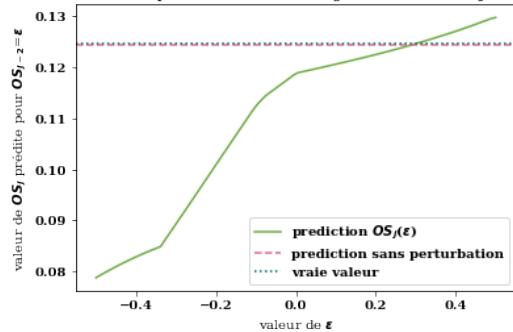


Effets d'une perturbation du montant des réserves en $j - 2$

Evolution de la prédiction de paiements par Deep triangle en j selon la perturbation des charges introduite en ($j - 2$)

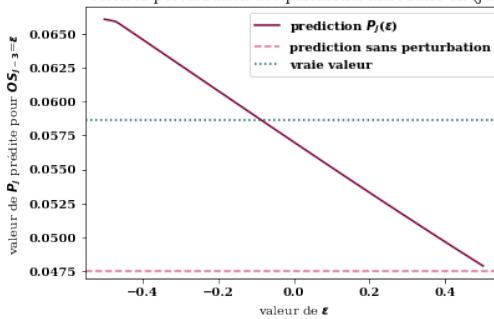


Evolution de la prédiction de réserve par Deep triangle en j selon la perturbation des charges introduite en ($j - 2$)



Effets d'une perturbation du montant des réserves en $j - 3$

Evolution de la prédiction de paiements par Deep triangle en j selon la perturbation des paiements introduite en ($j - 3$)



Evolution de la prédiction de réserve par Deep triangle en j selon la perturbation des paiements introduite en ($j - 3$)

