Prévision des tassements induits par le creusement au tunnelier : construction d'une base de données et apprentissage automatique

Tatiana Richa

École des Ponts ParisTech

THÈSE DE DOCTORAT

de l'École des Ponts ParisTech

Prévision des tassements induits par le creusement au tunnelier : construction d'une base de données et apprentissage automatique

École doctorale N°531, Sciences, Ingénierie et Environnement (SIE)

Spécialité : Géotechnique

Thèse préparée au laboratoire Navier et à Setec (terrasol et tpi)

Thèse soutenue le **10 mai 2023**, par
Tatiana RICHA

Composition du jury :

Jean, SULEM Professeur, École des Ponts ParisTech	*Président*
Pierre BREUL Professeur des universités, Polytech Clermont	*Rapporteur*
Daniel DIAS Professeur des universités, Université Grenoble Alpes	*Rapporteur*
Marie-Aurélie CHANUT Docteur, CEREMA	*Examinatrice*
Michel PRÉ Docteur, Setec tpi	*Examinateur*
Didier SUBRIN Docteur, Centre d'Etudes des Tunnels	*Examinateur*
Jean-Michel PEREIRA Professeur, École des Ponts ParisTech	*Directeur de thèse*
Lina-María GUAYACAN-CARRILLO Chargée de recherche, École des Ponts ParisTech	*Co-encadrante de thèse*
Gilles CHAPRON Directeur d'études, Setec terrasol	*Invité, encadrant entreprise*
Francis LANQUETTE Directeur d'études, Setec tpi	*Invité, encadrant entreprise*
Hervé LE BISSONNAIS Directeur délégué, Setec terrasol	*Invité*

Remerciements

Ce travail de thèse s'inscrit dans le cadre d'une convention CIFRE reliant les sociétés Setec terrasol et Setec tpi et le laboratoire Navier de l'Ecole des Ponts ParisTech. Ma profonde gratitude va en premier à Terrasol et sa direction, Valérie BERNHARDT, Hervé LE BISSONNAIS et Fahd CUIRA, pour m'avoir offert un excellent cadre pour réaliser cette expérience passionnante.

Je tiens à exprimer ma profonde gratitude envers les personnes qui ont suivi de près mes travaux de thèse et qui m'ont offert leurs précieux conseils ainsi que leur soutien indéfectible, notamment mon directeur de thèse, Jean-Michel PEREIRA, pour la confiance qu'il m'a accordée pour m'exprimer à travers mes travaux tout en garantissant une évolution robuste grâce à son expertise et sa bienveillance. Ces remerciements vont également à mon encadrant Gilles CHAPRON pour son regard pratique et sa rigueur scientifique ainsi qu'à mon encadrante Lina-María GUAYACAN-CARRILLO pour son accompagnement et ses conseils judicieux. Travailler avec une équipe d'encadrement aussi compétente a été un privilège. Leur accompagnement régulier a énormément contribué à l'amélioration de mes réflexions, du rendu de ce manuscrit, et surtout à mon épanouissement professionnel.

Je remercie également mon encadrant Francis LANQUETTE pour son regard pratique sur cette thèse, qui nous a permis de garder l'essentiel en vue. Mes remerciements vont également aux membres de mon comité de suivi, Martin CAHN, Hervé LE BISSONNAIS, Jean SULEM et Aric WIZENBERG, pour leurs commentaires constructifs et leurs contributions précieuses lors de nos réunions semestrielles. Leurs conseils ont grandement amélioré ce travail.

Je souhaite également remercier les membres du jury de thèse, Marie-Aurélie CHANUT, Michel PRÉ, Didier SUBRIN et Jean SULEM pour leur temps et leur évaluation approfondie de cette recherche. Des remerciements particuliers sont adressés à mes rapporteurs Pierre BREUL et Daniel DIAS pour la lecture du manuscrit ainsi que pour leurs précieux commentaires et suggestions sur divers aspects de cette thèse.

Je ne saurais oublier de remercier Selmane LEBADOUI pour sa contribution indéniable à cette thèse pendant son stage de fin d'étude en data science chez Terrasol. Je lui suis profondément reconnaissante. Je souhaite également remercier les personnes que j'ai rencontrées sur les chantiers de la ligne 14 Sud et 15 Sud-Ouest du Grand Paris Express qui ne se sont pas lassées de répondre à mes questions, notamment Anthony BACHELIER et Iris HERRERA MARTIN.

Je tiens à remercier chaleureusement mes collègues pour leur soutien constant tout au long de ces trois années, et tout particulièrement Hiba, Éléa, Stéphanie et Laëtitia, qui

m'ont offert leur soutien inconditionnel et leur amitié inébranlable. La bienveillance des « Terrasoliens » a transformé ce parcours exigeant en une aventure agréable riche en émotions. Je tiens également à remercier mes collègues de l'équipe CERMES du laboratoire Navier pour leur accueil chaleureux et les bons moments partagés ensemble.

J'exprime ma profonde gratitude envers ma famille et mes amis pour leur soutien, leurs encouragements et leur amour inconditionnels. Leur présence, même de loin, a été essentielle pour traverser les moments difficiles et mener à bien cette aventure académique.

Enfin, je souhaite remercier toutes les personnes qui ont croisé ma route pendant cette période et qui ont contribué à mon épanouissement personnel et professionnel. Votre présence a été précieuse et je vous en suis infiniment reconnaissante.

Résumé

Le développement urbain entraîne une forte croissance des infrastructures souterraines. Pour répondre à ces besoins, de nombreux projets de tunnels sont construits sous des zones fortement urbanisées, comme le Grand Paris Express qui est actuellement le plus grand projet de transport en Europe. Cependant, le creusement des tunnels induit des déformations en surface, qui peuvent entraîner des dommages aux structures environnantes. Pour déterminer et réduire la vulnérabilité des avoisinants, il est primordial d'améliorer les prévisions des déformations induites en surface lors des creusements. Exploiter à grande échelle les données tirées de l'auscultation des chantiers est un atout majeur pour y parvenir. Dans ce contexte, cette thèse présente le développement d'un outil permettant de prévoir les tassements au fur et à mesure du creusement à l'aide de données tirées de deux lignes du Grand Paris Express.

Les données produites lors d'un chantier de tunnel sont riches et de nature diverse : données d'auscultation (mesures de tassements en surface, en particulier), paramètres de pilotage du tunnelier (vitesse d'avancement, pression au front, pression et volume de mortier injecté à l'arrière de la jupe, etc.), paramètres géométriques du tracé et informations géologiques et géotechniques (stratigraphie, en particulier). Le premier défi consiste à extraire ces données brutes depuis les différentes sources et à les transformer en informations exploitables. Pour cela, des techniques de nettoyage sont utilisées pour réduire le bruit, supprimer les données aberrantes ainsi que gérer les données manquantes. Des techniques de lissage et de calage des formulations analytiques de tassement sur les mesures de tassements en surface sont également présentées. Ensuite, une base de données relationnelle est construite pour héberger les données. L'architecture de cette base a été conçue de façon à pouvoir exploiter les données spatiales, temporelles et leurs relations (l'avancement du tunnelier combine en effet ces deux composantes).

L'étape suivante consiste à mener une analyse exploratoire des données pour identifier des régularités et des liens potentiels entre les différents paramètres, et également détecter et traiter les éventuelles anomalies. Les paramètres ayant la plus grande influence sur la valeur des tassements sont alors sélectionnés. Des algorithmes d'apprentissage automatique sont utilisés pour prévoir les tassements induits en surface lors de l'excavation. Au total, six algorithmes sont testés et comparés : Linear Regression, Support Vector Machine, Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost) et Artificial Neural Networks. Les résultats de ces prévisions sont comparés aux mesures réelles des tassements. Des conclusions sont établies quant à l'intérêt et à la performance de chacune des méthodes de prévision mises en œuvre, ainsi que leur application pratique potentielle (extrapolation sur une même ligne du Grand Paris ou d'une ligne à l'autre).

Dans cette thèse, nous avons donc développé un outil opérationnel capable d'améliorer les prévisions au fur et à mesure de la collecte des données. La base de données constituée a servi de source pour alimenter des algorithmes d'apprentissage automatique et pourra être utilisée pour des applications ultérieures. Ce travail a permis de démontrer l'efficacité des algorithmes à base d'arbres pour la prévision du tassement maximal à l'axe du tunnel pour un entraînement au bout de quelques mois de creusement et à une distance de quelques centaines de mètres à l'avant du front du tunnel.

Mots-clefs : tunnelier, cuvette de tassement, intelligence artificielle, données d'auscultation, traitement des données, analyse des données

ABSTRACT

Urban development has led to a significant increase in the construction of underground infrastructures. To meet these needs, many tunnel projects are built in highly urbanized areas, such as the Grand Paris Express, which is currently the largest transportation project in Europe. However, tunneling induces surface settlements, which can cause damage to nearby structures. To assess and reduce the vulnerability of the surroundings, it is essential to improve predictions of surface deformations induced by the excavation. Exploiting data from site monitoring is a major asset in achieving this goal. In this context, this thesis describes how data from excavation of two lines of the Grand Paris Express were used to develop a tool that can adjust settlement predictions as excavation progresses.

Data produced during a tunneling project are rich and diverse, including monitoring data (surface settlement measurements in particular), tunnel boring machine (TBM) control parameters (advance rate, face pressure, pressure and volume of injected grout behind the shield, etc.), geometric parameters of the alignment, and geological and geotechnical information (especially stratigraphy). The first challenge is to extract this raw data from the different sources and transform it into usable information. To do this, data cleaning techniques are used to reduce noise, eliminate outliers, and manage missing data. Techniques for smoothing and fitting analytical formulations to surface settlement measurements are also presented. Then, a relational database is constructed. The architecture of this database is designed to exploit spatial and temporal data and their relationships (TBM advancement combines these two components).

The next step is to conduct exploratory data analysis to identify patterns and potential links between different parameters and to detect and treat any anomalies. The parameters with the greatest influence on the settlement values are then selected. Machine learning models are used to predict induced surface settlements during excavation. In total, six models are tested and compared : Linear Regression, Support Vector Machine, Decision Tree, Random Forest, Extreme Gradient Boosting (XGBoost), and Artificial Neural Networks. The results of these predictions are compared to the actual settlement measurements. Conclusions are drawn regarding the usefulness and performance of each of the implemented prediction methods, as well as their potential practical application (extrapolation on the same Grand Paris line or from one line to another).

In this thesis, we developed an operational tool that can improve the predictions as the excavation progresses. The constructed database performed as a reliable source to feed machine learning algorithms and can be used for future applications. This work has demonstrated the effectiveness of tree-based algorithms for predicting the maximum settlement at the tunnel axis for training after a few months of excavation and at a distance

of a few hundred meters in front of the tunnel face.

Keywords : tunnel boring machine, surface settlement, artificial intelligence, monitoring data, data processing, data analysis

Table des matières

Introduction Générale

Contexte

L'utilisation de l'espace souterrain est une solution contemporaine à la forte urbanisation et à la demande croissante de mobilité. En effet, les tunnels permettent de déployer des modes de déplacement écologiques, rapides et à faible impact dans les zones urbaines. De nombreux projets de tunnels sont ainsi construits sous des zones fortement urbanisées. C'est le cas du Grand Paris Express qui est actuellement le plus grand projet de transport en Europe.

Le principal risque lors des excavations souterraines est le mouvement induit à la surface qui peut endommager les structures avoisinantes. Une estimation précise de ces déformations est donc cruciale pour anticiper les dégradations potentielles. Cependant, les déformations en surface sont un phénomène complexe qui dépend de nombreux paramètres, dont la nature du sol, la méthode de creusement, et la géométrie du tunnel. La relation entre les déformations et ces paramètres n'est pas simple et linéaire, notamment à cause du comportement complexe du sol, qui est difficile à appréhender.

L'estimation des mouvements engendrés en surface préalablement au creusement est effectuée en suivant des approches que l'on peut qualifier de « traditionnelles », telles que les méthodes empiriques (formules mathématiques) ou numériques (méthode des éléments finis, par exemple). Ces dernières ont été perfectionnées au fil des années à travers de nombreuses études mais restent cependant limitées par la nature tridimensionnelle du creusement des tunnels ainsi que par les lois de comportement de sol développées jusqu'à ce jour. En effet, les résultats sont dépendants de la loi de comportement choisie pour reproduire la réponse du sol face à l'excavation. Néanmoins, il s'avère difficile d'arriver à reproduire l'ensemble des phénomènes observés dans le terrain.

Pour suivre l'évolution des déformations au cours de l'excavation, de nombreux capteurs sont installés en surface ainsi qu'en profondeur. Aujourd'hui, ces mesures servent à donner l'alerte lorsque des anomalies sont observées, afin de prendre des actions correctives en un délai acceptable, et éventuellement effectuer des recalages ponctuels par méthode numérique, recalages qui nécessitent du temps de spécialistes.

Nous nous intéressons dans cette étude aux mesures de tassements induits en surface par le creusement des tunnels. Ces mesures sont obtenues à travers une série de théodolites automatiques qui enregistrent les déplacements du sol et des bâtiments avec une fréquence de mesure élevée.

La surveillance des chantiers de tunnel génère en effet une grande quantité de données spatiales et temporelles. Ce sujet de thèse est né de la volonté de tirer partie de cette grande quantité de données à l'heure où le Big Data et l'Intelligence Artificielle se démocratisent pour énormément de cas d'usage. L'exploitation de ces données est potentiellement un atout majeur pour réduire les incertitudes sur les tassements induits par le creusement, optimiser le suivi du creusement des ouvrages souterrains, et, par

conséquent, leur conception. L'objectif est donc d'utiliser ces données pour réévaluer les prévisions de tassement sans passer par des recalages complexes, laborieux et gourmands en temps passé. La collecte, le traitement et l'analyse de ces données sont cependant un défi en soit. La transformation des données brutes en informations exploitables est une tâche souvent ardue et chronophage, mais qui, une fois le processus établi, peut être la plupart du temps automatisée.

Ensuite, se pose la question du stockage de la donnée dans un format facilement exploitable pour ses différentes utilisations. Là encore, revient souvent la thématique des bases de données, par exemple relationnelles, pour stocker efficacement les mesures de tassement, mais aussi les informations spatiales, les données géologiques et géotechniques et les paramètres de creusement. Ces systèmes de stockage garantissent l'intégrité et le maintien des relations (spatiales et temporelles) entre les nombreux paramètres.

L'analyse et l'exploitation de cette grande quantité de données, est un domaine relativement nouveau dans la pratique géotechnique, ouvre la porte, nous le disions, à l'utilisation de méthodes de prédiction à l'aide d'algorithmes d'apprentissage automatique (Machine Learning). Toutefois, l'utilisation des méthodes à base d'Intelligence Artificielle n'est pas une tâche facile : il est nécessaire de bien comprendre comment la multitude d'algorithmes disponibles permettrait de répondre au problème spécifiquement posé, et de quelle manière les optimiser. C'est à cette seule condition qu'il est possible d'éviter le piège de la boîte noire en utilisant les bonnes techniques pour rendre les résultats de ces algorithmes compréhensibles et interprétables.

L'utilisation de ces techniques pour prédire les tassements n'est pas une nouveauté puisque la première tentative date de 1998. Cependant, l'état de l'art manque de résolutions pratiques du problème tenant compte de la disponibilité de la donnée au fur et à mesure de l'avancement du creusement. Il manque aussi bien souvent une quantité de données suffisante pour effectuer des tests fiables permettant de confirmer l'exactitude des résultats.

Objectifs

Cette thèse, réalisée dans le cadre d'une convention CIFRE avec setec Terrasol et setec tpi du groupe Setec, a pour objectif d'évaluer l'intérêt de l'utilisation d'algorithmes d'apprentissage automatique pour prévoir les tassements induits par le creusement au tunnelier à pression de terre. La finalité pratique est de guider et donner des moyens aux ingénieurs chargés du suivi de futurs travaux d'améliorer et ajuster les prévisions en parallèle de la réalisation des mesures de terrain. Les données utilisées sont tirées d'une partie des lignes 14 Sud et 15 Sud-Ouest du projet du Grand Paris Express. Les travaux d'extraction, de traitement et de construction d'une base de données sont présentés en détails. Ensuite, des études statistiques sont effectuées pour tirer des enseignements à partir de la grande quantité de données mise à notre disposition. Enfin, on cherche à prévoir les tassements à l'aide d'algorithmes d'apprentissage automatique. Cette thèse

s'est concentrée majoritairement sur les algorithmes à base d'arbres de décision tels que Decision Tree Regressor, Random Forest Regressor et Extreme Gradient Boosting. D'autres algorithmes sont également étudiés en guise de comparaison de performance.
Plusieurs approches sont considérées pour la prévision des tassements :

1. La première consiste à diviser aléatoirement les données, sans prendre en compte l'aspect spatial et temporel de la progression du creusement. L'objectif est d'ajuster les paramètres intrinsèques aux modèles et d'estimer dans quelle mesure il leur est possible de sortir de leur zone d'apprentissage pour généraliser sur des combinaisons de paramètres non rencontrées.
2. La deuxième a pour but de mettre en évidence la quantité de données nécessaire pour obtenir des algorithmes performants.
3. La troisième cherche à tenir compte de l'aspect spatial du creusement : la division des données pour l'entraînement des algorithmes n'est plus effectuée aléatoirement. On cherche à éprouver la prévision des tassements sur des zones nouvelles, qu'elles s'approchent ou non de situations déjà rencontrées
4. La quatrième approche est une approche pratique qui tient compte de l'aspect spatio-temporel du problème. L'idée est de créer une preuve de concept, POC (Proof Of Concept) d'un algorithme d'apprentissage automatique capable de prévoir les tassements en temps réel afin de pouvoir intervenir de manière anticipée en cas de besoin.

Structure de la thèse

Les travaux de cette thèse sont présentés en 3 parties.

La première partie est une étude de l'état de l'art sur le sujet. On décrit d'abord le creusement des tunnels, notamment au tunnelier à pression de terre, ainsi que les équations de tassement induits par le creusement. Les méthodes « traditionnelles » de prévision des tassements et leurs limitations sont présentées afin de mettre en évidence l'intérêt d'appliquer des méthodes « innovantes » basée sur l'Intelligence Artificielle. Ensuite, on cherche à démystifier l'Intelligence Artificielle et les algorithmes d'apprentissage automatique. Le but est de motiver les ingénieurs à utiliser ce type d'approches après une bonne compréhension de leur fonctionnement. Enfin, on présente une description de l'état de l'art de la prévision des tassements à l'aide d'algorithmes d'apprentissage automatique. Ce dernier chapitre constitue également implicitement la méthodologie des travaux de cette thèse, à travers la description de la méthodologie adoptée dans les différentes études.

La deuxième partie décrit l'ingénierie des données du Grand Paris Express. On y présente une description détaillée des travaux de collecte, de nettoyage et de traitement des données. De plus, la construction de la base de données et son architecture sont

présentées.
Par la suite, on effectue une mise au point du problème à traiter afin de faire le bon choix des variables cibles (mesure à prévoir) et des caractéristiques (paramètres ayant une influence sur cette mesure). Des études statistiques détaillées sont effectuées et présentées.

La troisième partie est l'application des différentes approches de prévision des tassements à l'aide d'algorithmes d'apprentissage automatique. En premier lieu, on prévoit les tassements maximaux à l'axe du tunnel (s_{max}) et à une distance de l'axe du tunnel (s^*) avec une division aléatoire des données. Cet exercice sert à régulariser et à optimiser les algorithmes à utiliser ultérieurement. En second lieu, on avance une approche spatio-temporelle de prévision de s_{max}. En d'autre termes, on entraîne un algorithme en utilisant des données récoltées à l'arrière du front du tunnel afin de prévoir les tassements à l'avant du front.

Enfin, nous terminons ces travaux par des conclusions générales suivies de quelques perspectives qui serviront de points de départ pour des travaux ultérieurs sur le sujet. Des recommandations d'ordre pratique sur les données et sur la méthodologie d'application de l'apprentissage automatique sont également proposées.

Partie I

Étude Bibliographique

Introduction

La première partie de cette thèse est dédiée à l'étude bibliographique du sujet. Celle-ci est divisée en trois chapitres étant donné que cette étude s'intéresse à deux domaines : celui des déformations induites par le creusement des tunnels et celui de l'Intelligence Artificielle.

Le Chapitre 1 est consacré à l'étude bibliographique sur les méthodes de creusement, avec un intérêt particulier pour le tunnelier à pression de terre, justifié par le fait que les lignes de métro étudiées dans cette thèse sont creusées avec ce type de tunnelier. Ensuite, on rappelle les origines des déformations observées en surface ainsi que les équations du tassement. La notion de progression du tassement avec l'avancement du creusement est introduite. L'intérêt est de décrire ainsi les mesures en surface obtenues pour une cible visée de façon régulière dans le temps. La dernière partie se concentre sur la description des méthodes actuelles de prévision du tassement ainsi que les limites de ces techniques. L'intérêt de tester des méthodes innovantes telles que l'application d'algorithmes à base d'Intelligence Artificielle est ainsi mis en relief.

Le Chapitre 2 est une introduction au monde de l'Intelligence Artificielle, le but étant de démystifier les terminologies et les méthodes d'apprentissage automatique.

Le Chapitre 3 constitue à la fois une description de l'état de l'art de la prévision des tassements à l'aide d'algorithmes d'apprentissage automatique, et une introduction à la méthodologie de travail adoptée dans le cadre de cette thèse à travers la description des différentes étapes d'un exercice d'apprentissage automatique.

TASSEMENTS INDUITS PAR LE CREUSEMENT DES TUNNELS

1

Introduction

La réalisation d'ouvrages souterrains est aujourd'hui un enjeu prioritaire dans le cadre des politiques de (ré)aménagement urbain, d'amélioration de la viabilité et du développement des transports en commun. Pour répondre à ces besoins, les techniques de creusement des tunnels évoluent depuis le $XIX^{ème}$ siècle pour s'adapter en particulier aux enjeux de creusement en milieu urbain dense, aux natures des terrains rencontrés et aux contraintes de temps. En effet, le creusement d'un tunnel perturbe le champ initial des contraintes dans le terrain ainsi que les conditions hydrogéologiques, et cela, quelle que soit la technique utilisée. Cette modification des contraintes s'accompagne de mouvements des terrains qui sont sources de risques notamment dans les milieux urbains où les déformations peuvent nuire aux structures existantes en surface et en profondeur. Pour garantir la sécurité des avoisinants, il faut pouvoir prédire précisément ces mouvements. De nombreuses méthodes existent pour y parvenir et seront détaillées dans la suite. Mais ces estimations sont inévitablement empruntes d'aléas, le sol étant par nature complexe à qualifier. La réduction des incertitudes sur les prévisions des tassements induits en surface est donc une motivation récurrente et fondamentale. L'estimation des mouvements engendrés en surface préalablement au creusement est effectuée par des approches empiriques, analytiques, numériques et, plus récemment, à l'aide d'algorithmes d'Intelligence Artificielle.
Ce chapitre présente l'historique des méthodes de creusement des tunnels. On détaille particulièrement les tunneliers à pression de terre, puisque les tunnels étudiés dans cette thèse sont creusés par ce type de tunnelier. Ensuite, on examine les déformations observées en surface : leurs origines et leur description mathématique tridimensionnelle. La dernière partie décrit les méthodes de prévision du tassement ainsi que les avantages et inconvénients de chaque approche.

1.1 Méthodes de creusement mécanisées : les tunneliers

Le catalogue des méthodes de creusement des tunnels est vaste et en perpétuelle évolution, s'étendant des méthodes traditionnelles comme l'explosif pour les tunnels au rocher et les méthodes de creusement mécanisé avec différents types de tunnelier. Cette partie est une introduction historique aux creusement des tunnels et décrit plus en détail le creusement au tunnelier à pression de terre (EPB) et ses composants.

1.1.1 Historique des tunnels : des méthodes traditionnelles au creusement au tunnelier

Méthodes traditionnelles

Les premières exploitations du sous-sol datent au moins de la fin du néolithique en Europe avec les galeries pour la recherche de minéraux variés. Un des exploits de l'ingénierie antique est le tunnel d'Eupalinos en Grèce avec ses 1 036 m de long. Sans remonter jusqu'à l'antiquité, on peut dire que le premier tunnel « moderne » construit pour une infrastructure est celui de Terrenoire sur la ligne de Roanne à Andrézieux, France (PlanèteTP, 2022 ; Waldmann, 2005). Le $XIX^{ème}$ siècle a connu le commencement de la construction de longs tunnels comme Blaisy (4 100 m en 1846) entre Paris et Dijon ou le tunnel de Saint Clair (2 403 m) achevé en 1890. L'année 1900 fut l'année de construction de la première ligne du métro Parisien, le septième du monde après ceux de Londres, New-York, Chicago, Budapest, Glasgow et Vienne. La France et l'Italie construisent entre 1955 et 1962 le tunnel du Mont Blanc, le plus long du monde avec 11 600 m et le premier à passer sous un massif montagneux, le plus haut d'Europe. Ces deux pays ont battu leur propre record avec le tunnel du Fréjus de longueur 12 868 m, construit entre 1974 et 1980.

Au $XIX^{ème}$ siècle, l'excavation se faisait par section divisée avec un soutènement en bois. Les historiens des tunnels différencient les méthodes selon la division de la section (autrichienne (NATM), belge, allemande). Au début du $XX^{ème}$ siècle, les techniques de forage se perfectionnent ainsi que les techniques de traitement des terrains (Guilloux et al., 2021). Les soutènements en cintres métalliques, boulons et béton projeté ont donné lieu à de nouvelles méthodes d'excavation en pleine section ou en section divisée.

Méthodes mécanisées

Le tunnelier, souvent nommé TBM (de l'anglais Tunnel Boring Machine) est une machine permettant une excavation entièrement mécanisée. Les premières machines sont apparues dans les années 50 avec les tunneliers de type gripper et se sont perfectionnées en permettant notamment de confiner le terrain en appliquant une pression sur le front d'excavation avec de l'air, de la terre ou de la boue. Il existe aussi des tunneliers bi-mode (on peut basculer d'une pression de terre à une pression de boue) et à densité variable (AFTES, 2019). En 2019, environ 75% des tunneliers utilisés sont des tunneliers à confinement dont 42% sont des tunneliers à pression de terre (EPB) et 23% des tunneliers à pression de boue (AFTES, 2019) (Figure 1.1). La gamme de diamètres des machines la plus courante est de 8 à 10 m, ce qui correspond aux tunnels réalisés pour les infrastructures de transport (métros à double voie ou tunnels routiers comme le duplex A86 à Paris (Terrasol, 2023 ; Vinci, 2023)). Les principaux fabriquants de tunneliers sont allemands (Herrenknecht), américains (The Robbins Company) et chinois (China Railway Engineering Equipment Group). Il existe également sur le marché des fabriquants français, comme Bessac.

Le tunnelier assure simultanément les fonctions suivantes :

1. Le creusement et le marinage des déblais
2. Le soutènement du front de taille (préserver la stabilité du front)
3. La pose du revêtement à l'arrière du bouclier

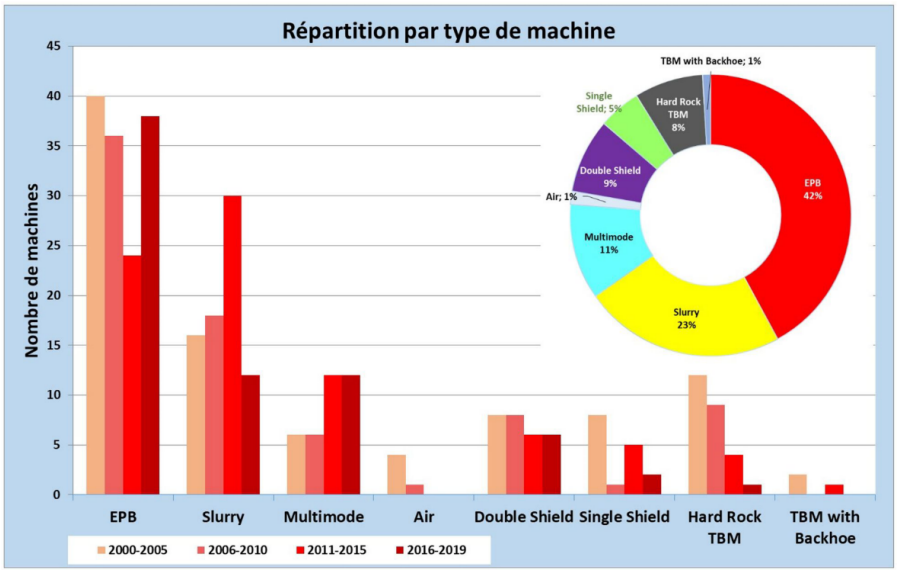


Figure 1.1. Répartition de l'utilisation des types de tunneliers selon les périodes (AFTES, 2019)

1.1.2 Tunnelier à pression de terre

L'intérêt particulier porté dans ce chapitre pour le tunnelier à pression de terre se justifie par le fait que les lignes de métro étudiées dans cette thèse sont creusées avec ce type de tunnelier.
Le développement des tunneliers à pression de terre (EPB) a commencé en 1974 avec la construction d'un réseau de drainage des eaux pluviales à Tokyo. La première utilisation d'un EPB pour un tunnel d'une infrastructure de transport date de 1986 en Essen, Allemagne (Herrenknecht et al., 2011).

Composants d'un tunnelier à pression de terre

Un tunnelier à pression de terre est composé de plusieurs éléments assurant chacun une fonction distincte. Des capteurs mesurent une grande variété de paramètres pour chaque élément afin de surveiller son fonctionnement. Les composants (présentés également dans la Figure 1.2) et les paramètres principaux d'un EPB sont décrits ci-dessous :

Roue de coupe (RDC) : composant frontal du tunnelier au contact du sol sur lequel sont fixés les outils de coupe (molettes et dents) (Figure 1.6g). Les molettes se comportent comme un couteau pour excaver le sol et doivent être adaptées à la

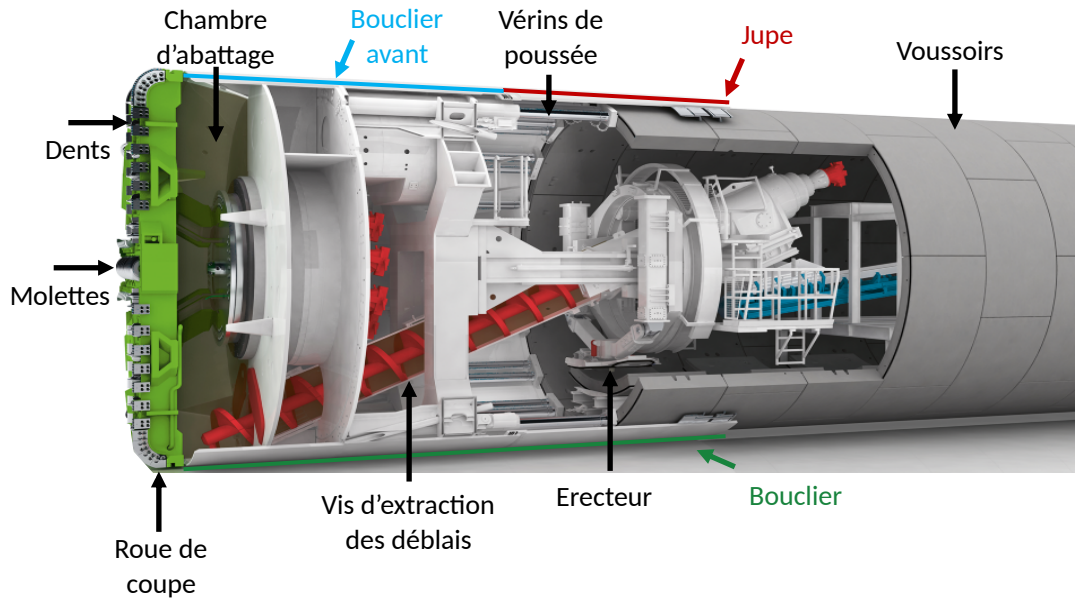


Figure 1.2. Composants principaux d'un tunnelier à pression de terre (adaptée de Herrenknecht, 2022)

nature des terrains au front (terrains durs ou tendres). Les paramètres mesurés sont, entre autres :

- Vitesse d'avancement du tunnelier : $V_{tunnelier}$ [mm/min]
- Vitesse de rotation de la RDC : V_{RDC} [tour/min]
- Couple de rotation (moment) de la RDC : M_{RDC} [kN.m]
- Puissance de la RDC : P_{RDC} [kW]
- Énergie de la RDC : E_{RDC} [mJ/m^3]
- Longueur excavée pour un anneau : L_{RDC} [mm]
- Poussée exercée par le terrain au front sur les molettes : $P_{molettes}$ [kN]
- Date de début et de fin d'excavation ainsi que la durée d'excavation

Chambre d'abattage : le sol excavé est conservé dans la chambre d'abattage où il est mélangé avec de l'eau et des additifs appropriés (mousses ou polymères) pour former une pâte homogène. Cela permet au sol d'atteindre un état de plastification et de lubrification suffisant pour pouvoir être extrait correctement à l'aide de la vis d'Archimède. La pâte formée est mise sous pression dans la chambre d'abattage pour appliquer une pression au front. Cette dernière est également exercée avec de l'air mis sous pression en partie supérieure de la chambre d'abattage. La pression au front doit être a minima égale à la pression d'eau dans le massif et a maxima égale à la pression de soulèvement au dessus du tunnel (Figure 1.3). Elle est mesurée par 12 capteurs répartis sur l'ensemble du pourtour de la chambre d'abattage (Figure 1.4). Les paramètres mesurés sont, entre autres :

- Pression au front, mesurée par le capteur placé en voûte du tunnel : P_{front} [bar]

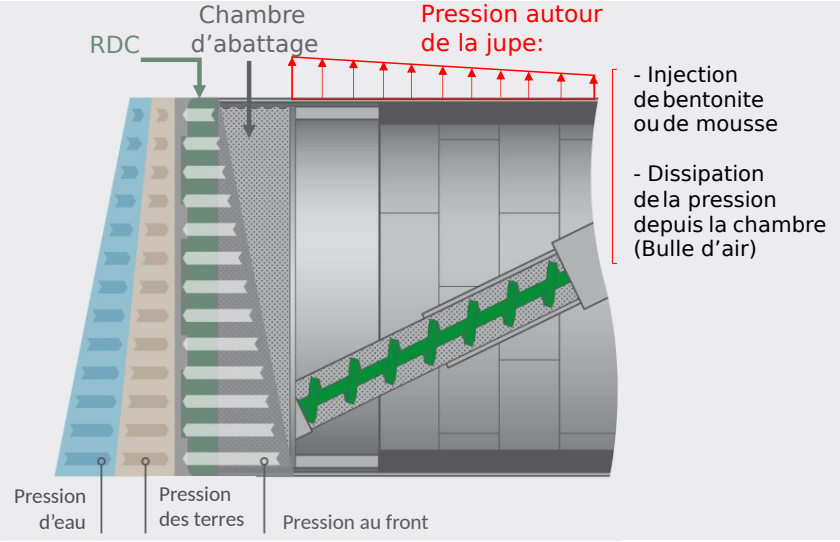


Figure 1.3. Stabilité du front assurée par la pression au front (adaptée de Herrenknecht, 2022)

– Taux de remplissage de la chambre qui informe sur le volume de sol à l'intérieur de la chambre d'abattage : $\tau_{remplissage}$ [%]
– Volume d'eau ajoutée : V_{eau} [m^3]
– Volume d'additifs ajoutés : $V_{polymères}$ [m^3]

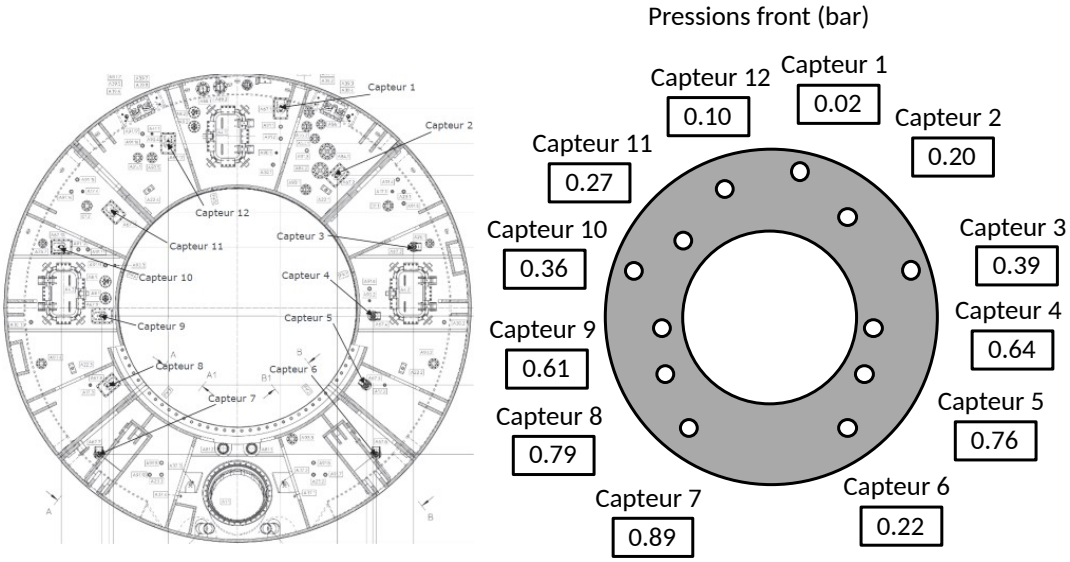


Figure 1.4. Répartition des 12 capteurs dans la chambre d'abattage pour mesurer la pression au front (document interne à Terrasol, Setec)

Vis d'extraction : également appelée vis d'Archimède ou vis sans fin, elle permet l'extraction des matériaux d'abattage remplissant la chambre pour les déposer sur le tapis convoyeur. La quantité de matériau retiré est déterminée par la vitesse de rotation de la vis. Le contrôle du volume de matériau extrait par rapport à la vitesse d'avancement du tunnelier est primordial dans la maîtrise du confinement du front de taille. La vis a aussi pour rôle d'absorber la différence de pression entre la RDC

et la pression atmosphérique (arrière de la vis). La vitesse de rotation de la vis est contrôlée et le débit est asservi pour maintenir une pression constante au front. Les paramètres mesurés sont :

- vitesse de rotation de la vis : V_{vis} [tour/min]
- couple de rotation (moment) de la vis : M_{vis} [kN.m]
- pression des terres au milieu de la vis : P_{vis} [bar]
- ratio vitesse de la vis sur vitesse d'avancement du tunnelier : $R_{vis-avancement}$

Convoyeur à bandes : (ou tapis convoyeur) tapis roulant qui transporte les déblais depuis leur extraction par la vis jusqu'à l'extérieur (Figure 1.6c et Figure 1.6d). Des pesons sont placés au début et à la fin du tapis pour peser la masse de déblais extraite. Les principaux paramètres mesurés sont :

- Poids brut du 1^{er} peson : P_{peson1} [ton]
- Poids brut du 2^{nd} peson : P_{peson2} [ton]

Voussoir : élément préfabriqué en béton armé (Figure 1.6a), posé sur le pourtour du tunnel à l'abri de la jupe à l'aide d'un érecteur (Figure 1.6k). Le revêtement final du tunnel est constitué d'anneaux, eux-mêmes composés de 7 voussoirs chacun pour les lignes 14 Sud et 15 Sud-Ouest du Grand Paris Express. Les anneaux sont munis d'un dispositif d'étanchéité. Pour dimensionner les voussoirs, il faut tenir compte des charges statiques de pression du terrain et de l'eau en plus des efforts de poussée des vérins. Les paramètres enregistrés lors du creusement concernant les voussoirs sont :

- Date de début de pose des voussoirs
- Date de fin de pose des voussoirs
- Durée de pose de l'anneau

Érecteur : manipulateur télécommandé pour positionner les voussoirs pendant la construction de l'anneau (Figure 1.6k).

Vérins de poussée : le tunnelier avance à l'aide de vérins qui s'appuient sur les voussoirs et poussent ainsi le tunnelier vers l'avant (Figure 1.6l). La force de poussée sur les voussoirs étant importante et concentrée, elle conditionne le dimensionnement des voussoirs. Un des paramètres mesurés est :

- Pression de poussée d'un groupe de vérins (6 groupes au total) : $P_{vérins}$ [bar]

Bouclier et Jupe : le bouclier est un cylindre métallique qui garantit la protection et l'étanchéité du travail d'excavation sur le front de taille. Il se termine par une « jupe » à l'abri de laquelle sont mis en place les voussoirs composant le revêtement du tunnel (Figure 1.5). La pression autour de la jupe est causée d'une part par la dissipation de la pression de la chambre d'abattage et d'autre part par des injections de bentonite ponctuelles en cas de besoin autour de la jupe et du bouclier (Figure 1.3).

A l'avant de la jupe est placé le **fontimètre** qui est un outil de mesure de la distance terrain-bouclier. Cette mesure renseigne sur la sur-excavation ou le déconfinement du sol après l'excavation.
A l'arrière de la jupe, des **brosses** assurent l'étanchéité entre l'intérieur de la jupe et l'extérieur du revêtement.
La surcoupe de la roue de coupe, et la conicité de la jupe (différence entre le diamètre de creusement et le diamètre à la fin de la jupe) induisent un **vide annulaire** à l'arrière de la jupe. Ce vide est source de convergence du terrain et donc de tassements. Il doit donc être comblé rapidement après la pose des voussoirs. Pour cela, on injecte du mortier de bourrage à travers des cannes d'injection situées à l'arrière de la jupe. Le mortier, alimenté par le système d'injection (Figure 1.6p), est mis en place sous pression à la sortie du joint de queue de la jupe. Les lignes d'injection (Figure 1.6l) en service sont équipées d'un capteur de pression situé au plus près de la jupe. Par conséquent, la pression d'injection mesurée est plus grande que la pression reçue par le massif. Les brosses à l'arrière de la jupe permettent d'éviter (sauf cas accidentel) la migration du mortier vers l'avant de la machine. les paramètres mesurés sont notamment :

- Angles de guidage horizontal et vertical du tunnelier : G_H [mm] et G_V [mm], respectivement
- Pression moyenne d'injection du mortier de bourrage calculée à partir des pressions d'injection des différentes cannes autour du tunnel : $P_{mortier}$ [bar]
- Volume total d'injection de mortier : $V_{mortier}$ [m^3]
- Volume théorique d'injection de mortier : $V_{mortier\ théorique}$ [m^3]
- Nombre total de coups de pompe de mortier : $N_{coups\ mortier}$

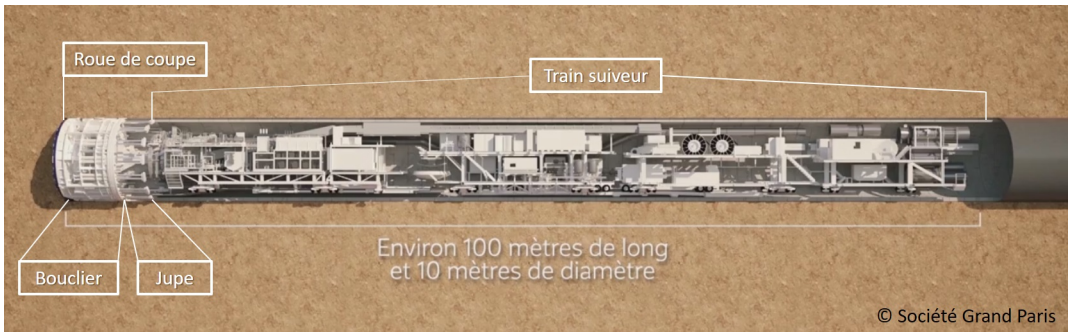


Figure 1.5. Eléments d'un tunnelier (SociétéGrandParis, 2021)

Train suiveur : d'environ 100 m de long, le train suiveur assure la logistique tunnelier (Figure 1.5). Il porte les équipements de puissance et de commande de la machine ainsi que les équipements nécessaires à la manutention des voussoirs, à la mise en oeuvre du mortier de bourrage et au marinage. Il permet également le transfert entre la logistique d'approvisionnement du tunnel et la machine (réseaux, puissance, marinage, eaux, air comprimé, ...)

Le fonctionnement d'un tunnelier à pression de terre est le suivant : la machine creuse d'une longueur égale à celle des voussoirs (soit environ 2 mètres) avant de s'arrêter. A ce stade, l'érecteur pose les voussoirs pour former un anneau (7 voussoirs dans le cas des lignes 14 Sud et 15 Sud-Ouest du Grand Paris Express) . Les vérins peuvent ainsi s'appuyer sur le nouvel anneau et le tunnelier commence le creusement de l'anneau suivant. Si N est le dernier anneau posé, l'injection de boue se fait à l'arrière de l'anneau N-1 lors du creusement de l'anneau N+1. Pour une jupe de 10 m et des anneaux de 2 m, l'injection de mortier se fait 14 m derrière la RDC.
Une série de photos prises d'un tronçon de la Ligne 15 Sud-Ouest du projet du Grand Paris Express est proposée en guise d'exemple d'un tunnelier à pression de terre (Figure 1.6).

(a) Voussoirs en stock et en cours de transport

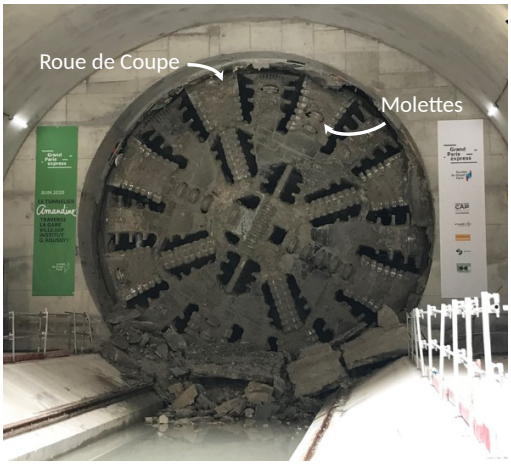
(b) Vue de l'extérieur du tunnel

(c) Convoyeur à bande transportant les déblais de la chambre d'abattage vers l'extérieur du tunnel

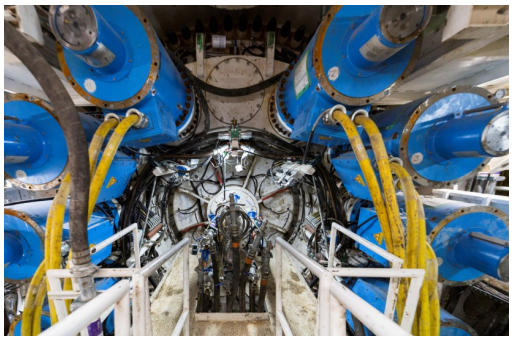
(d) Bac des déblais à la sortie du tunnel

(e) Cabine de pilotage du tunnelier

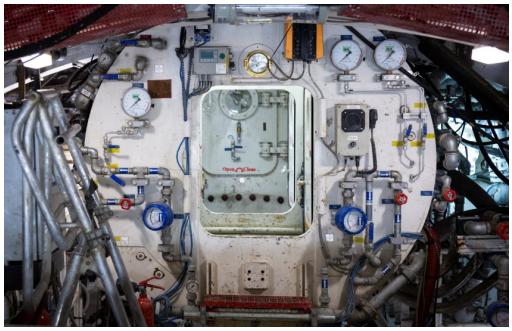
(f) Chambre de survie en cas d'accident



(g) Roue de Coupe et Molettes

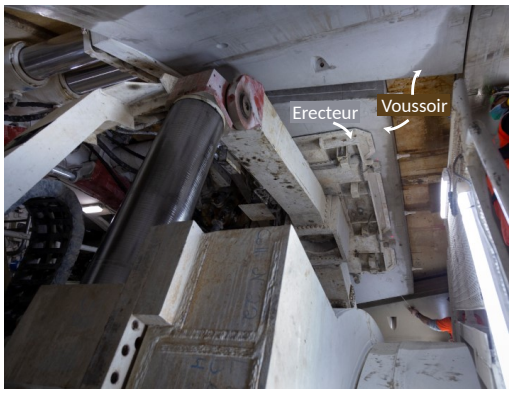


(h) Moteurs de rotation de la Roue de Coupe

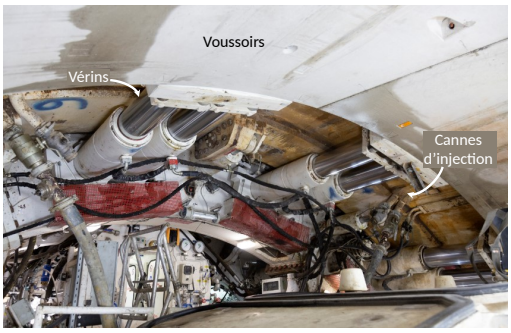


(i) Chambre hyperbare (avant l'entrée dans la chambre d'abattage)

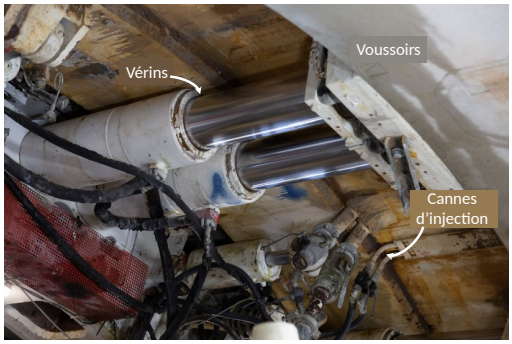
(j) Prise en charge du voussoir à l'intérieur du tunnel



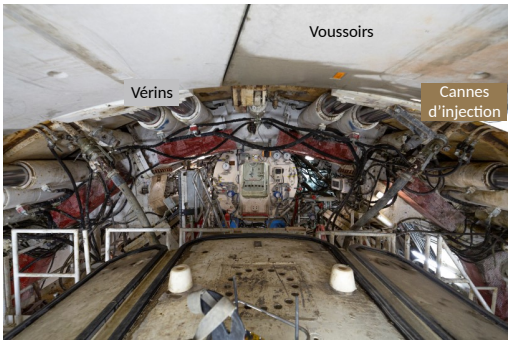
(k) Mise en place du voussoir à l'aide de l'érecteur



(l) Vérins de poussée et cannes d'injection du mortier de bourrage derrière les anneaux



(m) Vue de proche des vérins de poussée et cannes d'injection du mortier de bourrage derrière les anneaux



(n) Vue globale des vérins de poussée et cannes d'injection du mortier de bourrage derrière les anneaux

(o) Usine de fabrication de mortier au chantier

(p) Pompe de mortier à l'intérieur du tunnelier

Figure 1.6. Image du tunnelier Amandine sortant de la gare Villejuif Institut Gustave Roussy (tronçon TR2 de la Ligne 15 Sud-Ouest du projet du Grand Paris Express) (Dargham, 2021)

1.2 Tassements induits en surface

Le creusement des tunnels est source de déformations en surface. Ces déformations sont susceptibles d'induire des dommages aux avoisinants (bâtiments, réseaux, ouvrages d'art). L'estimation préalable des tassements permet d'évaluer les paramètres de creusement nécessaires pour réduire les risques de dommages. Sont définies dans cette partie les principales origines des déformations ainsi que les relations empiriques développées par Peck (1969) et Attewell et Woodman (1982) pour prédire les tassements.

1.2.1 Sources de déformations

Le creusement des tunnels au tunnelier induit des déformations en surface. Généralement, ces déformations commencent à partir d'une certaine distance en avant du front, se poursuivent jusqu'à la pose du revêtement, et progressent parfois consécutivement à la pose. Les sources de tassement liées aux travaux de creusement au tunnelier peuvent être décomposées en 5 catégories (AFTES, 1995 ; Mair et Taylor, 1997) (Figure 1.7) :

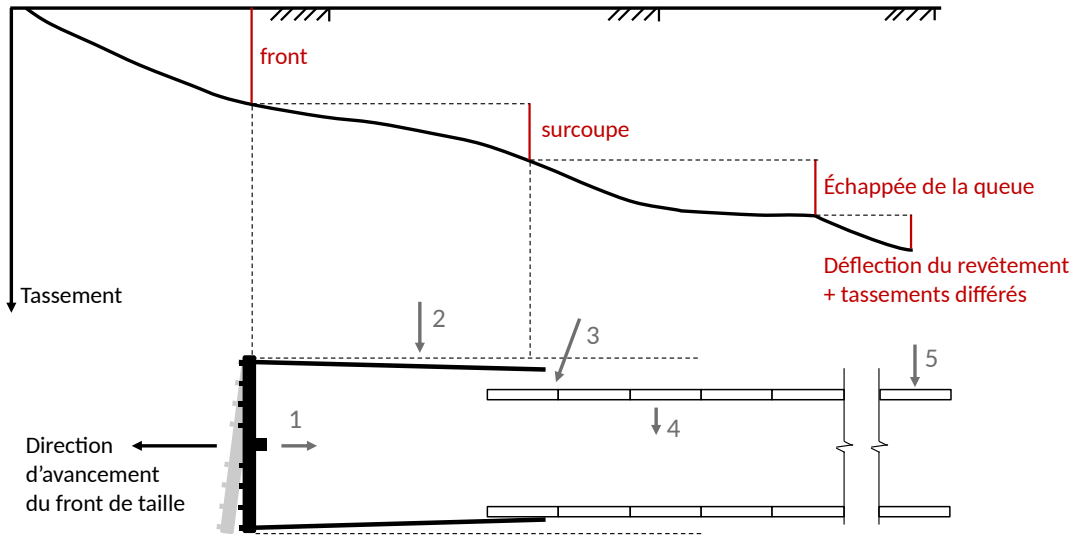


Figure 1.7. Principales sources de déformations lors du creusement au tunnelier et évolution des tassements le long de l'axe d'excavation (adapté de (AFTES, 1995 ; Moller, 2006))

1. **Extrusion :** Mouvements du terrain vers le front de taille dus au déconfinement du sol. L'amplitude de l'extrusion dépend de la qualité du confinement dans la chambre d'abattage, de la nature des terrains et des conditions hydrauliques.
2. **Ensemble des déformations autour du bouclier et de la jupe :** Tassements au passage du front et le long du bouclier dûs notamment aux déformations induites. Celles-ci sont liées à la surcoupe, à la conicité de la jupe (différence entre le diamètre de creusement et le diamètre en fin de jupe, le guidage (angle d'inclinaison du guidage) et la réduction des contraintes tangentielles (glissement sol-bouclier).

3. **Vide annulaire :** Tassements induits par l'espace résiduel entre le terrain et l'extrados des voussoirs à l'échappée de la queue du bouclier. L'amplitude de ces mouvements dépend de la qualité du mortier de bourrage et de son injection.
4. **Déformation du revêtement :** Tassements dus à la déformation des voussoirs causée par la poussée du sol, la poussée des vérins ou encore la consolidation du mortier de bourrage injecté derrière les anneaux. Le fluage du béton à long terme peut également être source de déformations.
5. **Consolidation :** Tassements dus à la consolidation du massif (dissipation des surpressions interstitielles éventuellement générées lors du creusement).

1.2.2 Relations empiriques du tassement

Le tassement en surface en un point donné évolue en fonction de la progression de l'excavation, qu'elle soit exprimée en fonction du temps ou de la position du front d'excavation. La cuvette tridimensionnelle (s_{3D}) s'exprime, pour une position donnée du front du tunnel, comme la combinaison de l'expression des tassements dans les deux directions orthogonales : le tassement transversal (s_{trans}), fonction de la distance à l'axe du tunnel, et le tassement longitudinal (s_{long}), fonction de la distance au front du tunnel. Cette description établit donc la variation des tassements selon les coordonnées de l'espace plutôt que leur évolution dans le temps. Cette partie a pour objectif de rappeler les équations décrivant les tassements s_{trans}, s_{long} et s_{3D} pour ensuite introduire l'équation de progression du tassement avec l'avancement du creusement.

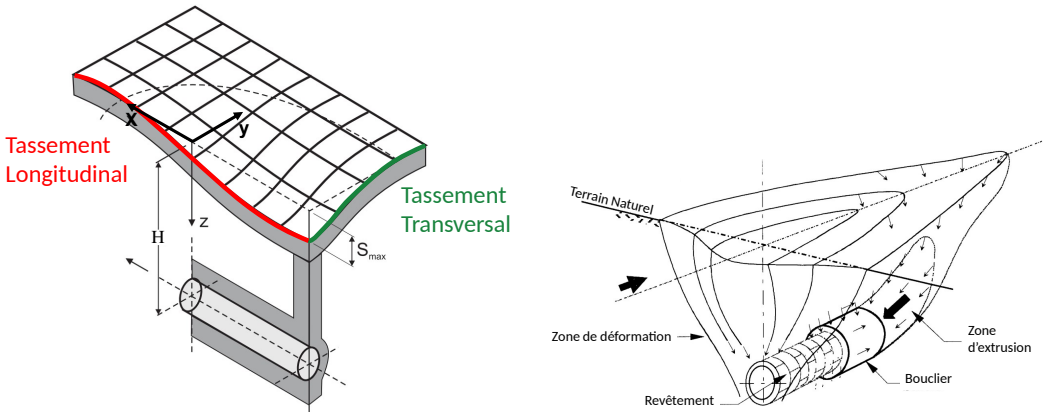


Figure 1.8. Cuvette tridimensionnelle observée au passage du tunnelier.
(a) adapté de (AFTES, 1995 ; Moller, 2006)
(b) adapté de (Suwansawat, 2002)

Notations et signes

Il convient tout d'abord de préciser les conventions de notations et de signe adoptées dans ce travail (Figure 1.9) :

– x : coordonnée curviligne le long de l'axe du tunnel. Le creusement se fait dans le sens positif de x.
– x_f : position du front du tunnel.
L'expression « à l'arrière du front » est équivalente à $x < x_f$ alors que « à l'avant du front » est équivalente à $x > x_f$
– d_{front} : distance au front du tunnel, $x_f - x$
– y : coordonnée dans la direction orthogonale à l'axe du tunnel, positif de façon à ce que (O, x, y, z) soit orthonormé.
– z : coordonnées dans la profondeur, positif vers le bas.
– d_{axe} : distance à l'axe du tunnel.

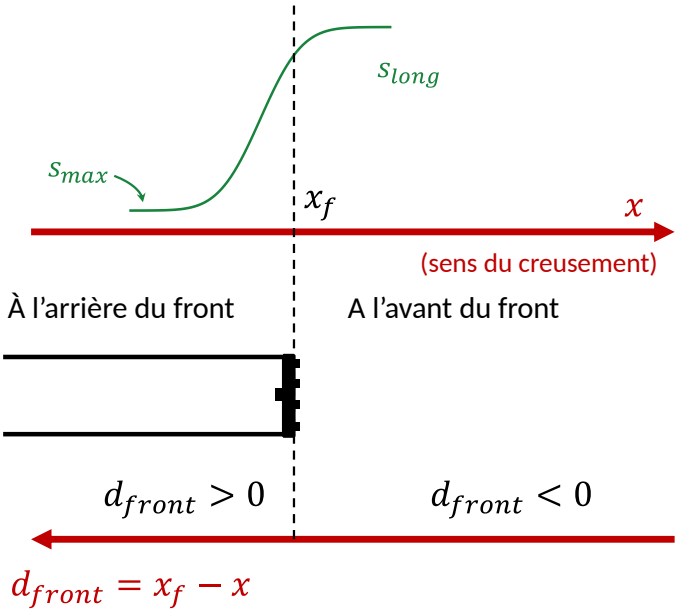


Figure 1.9. Conventions de signe selon l'avancement du tunnel et définitions de s_{max}, s_{long}, d_{front}, x_f

Tassement transversal

La courbe de tassement transversal observée loin à l'arrière du front du tunnel est exprimée par Peck (1969) comme ayant la forme d'une distribution gaussienne (Figure 1.10 et Figure 1.8) centrée sur l'axe du tunnel. Le tassement est donc maximum à l'axe, noté (s_{max}). La distance entre l'axe et les points d'inflexion de cette courbe transversale est notée (i_y). L'équation s'écrit :

$$s^*(y) = s_{max} \cdot \exp\left[-\frac{1}{2}\left(\frac{y}{i_y}\right)^2\right] \tag{1.1}$$

où s^* désigne le tassement maximal observé à une distance à l'axe égale à y.

Le volume de la cuvette de tassement V_s est évalué en calculant l'intégrale de l'Équation 1.1 :

$$V_s = \int s^*(y)dy = \sqrt{2\pi} \cdot i_y \cdot s_{max} \tag{1.2}$$

O'Reilly et New (1982) ont proposé d'exprimer i_y comme étant proportionnel à la profondeur du tunnel H [m] et donc indépendante de la méthode de creusement et du diamètre D du tunnel. On écrit donc :

$$i_y = k \cdot H \tag{1.3}$$

où k est une constante adimensionnelle qui dépend seulement du type de sol. Il rend compte de la façon dont les déformations profondes se propagent latéralement vers la surface. Ce paramètre est habituellement calé à l'aide d'études numériques ou bien déduit en fonction de la nature du sol au front et de la nature du massif à l'aide d'abaques (Guglielmetti et al., 2008, p. 146). A titre d'exemple, (Mair et Taylor, 1997) proposent des valeurs de $k = 0.5$ pour des tunnels dans des argiles et $k = 0.25$ pour des tunnels dans des sables et des roches.

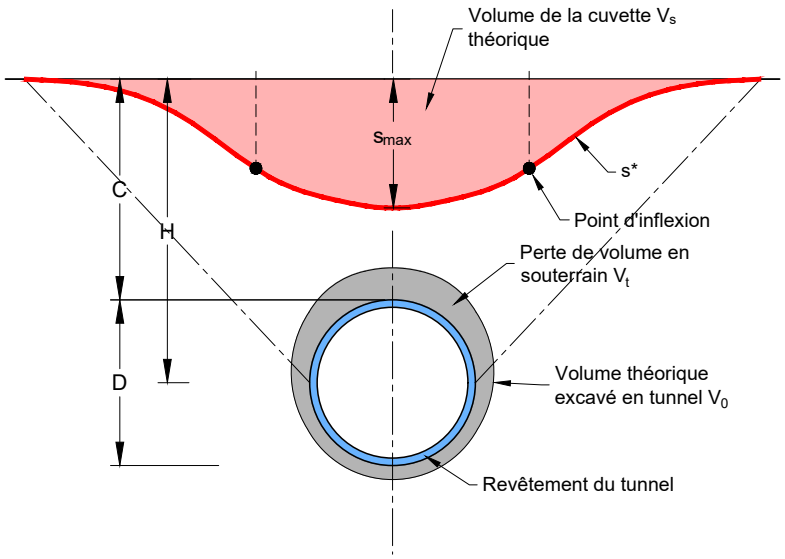


Figure 1.10. Cuvette de tassement de Peck (1969) (adaptée de (Guilloux et al., 2021))

Tassement longitudinal

La courbe de tassement longitudinal à l'axe du tunnel s_{long} représente les valeurs de tassement observées à l'avant et à l'arrière du front (Figure 1.8 et Figure 1.9). Attewell et Woodman (1982) l'ont exprimé comme ayant la forme d'une courbe gaussienne cumulée dont l'équation peut être écrite sous la forme suivante :

$$s(x) = s_{max} \cdot \left[G\left(\frac{x - x_i}{i_x}\right) - G\left(\frac{x - x_f}{i_x}\right) \right] \tag{1.4}$$

où $s(x)$ est le tassement observé à une distance $x_f - x$ du front du tunnel, x_i la position de départ du creusement, x_f la position du front du tunnel, i_x est un facteur de forme qui, comme dans l'Équation 1.1, rend compte de la position du point d'inflexion de la dérivée de cette gaussienne cumuléeet $G(\alpha)$ est la fonction de répartition de la loi Normale centrée réduite (moyenne nulle et écart-type égal à 1), dont la formule est :

$$G(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\alpha} \exp\left(-\frac{1}{2}\, x^2\right) dx \tag{1.5}$$

Par conséquent, si $\frac{x-x_f}{i_x}$ suit une loi Normale centrée réduite, on peut dire que x suit une loi Normale de moyenne x_f et d'écart-type i. Il convient de rappeler que la fonction de répartition d'une loi Normale de moyenne μ et d'écart-type σ s'écrit :

$$G_{[\mu,\sigma]}(\alpha) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\alpha} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] dx \tag{1.6}$$

Pour le creusement des tunnels au tunnelier, on considère que le tunnel est semi-infini. Par conséquent, la valeur de $x_i \longrightarrow -\infty$ et donc $G(\frac{x-x_i}{i}) \longrightarrow 1$. La formule du tassement longitudinal peut donc être simplifiée comme suit :

$$s(x) = s_{max} \cdot \left[1 - G_{[x_f, i_x]}(x)\right] \tag{1.7}$$

Cette expression tire sont origine d'observations réalisées par Attewell et Woodman (1982) pour un tunnel creusé avec un tunnelier en mode ouvert dans des argiles raides (Mair et Taylor, 1997). Ils ont constaté que, dans ces argiles raides, l'amplitude du tassement en surface à l'aplomb du front de taille est de l'ordre de $0.5\ s_{max}$. Dans le cas du creusement au tunnelier, le confinement limite fortement les tassements qui se développent à l'avant du front. Les observations consignées par (Mair et Taylor, 1997) pour des creusements au tunnelier à pression de terre réduisent cet ordre de grandeur à des valeurs autour de 0.25 à $0.3\ s_{max}$. En pratique, cela se traduit simplement par l'ajout à l'équation d'un paramètre m positif permettant de translater la courbe longitudinale selon x (Figure 1.11) :

$$\begin{aligned} s(x) &= s_{max} \cdot \left[1 - G_{[x_f - m, i_x]}(x)\right] \\ &= s_{max} \cdot \left[1 - G_{[-m, i_x]}(x - x_f)\right] \\ &= s_{max} \cdot G_{[m, i_x]}(x_f - x) \\ &= s_{max} \cdot G_{[m, i_x]}(d_{front}) \end{aligned} \tag{1.8}$$

La transformation de cette équation est possible compte tenu des caractéristiques du calcul intégral des courbes gaussiennes.

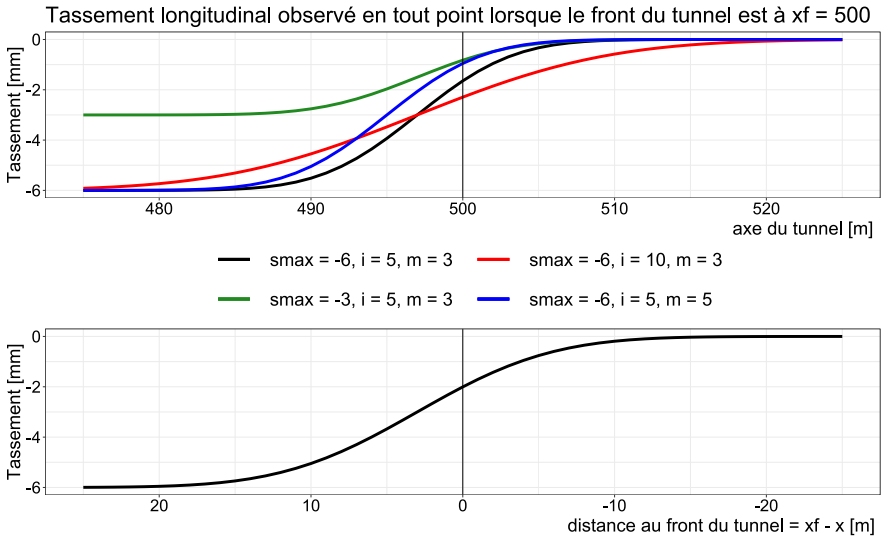


Figure 1.11. Illustration de l'influence des paramètres sur la forme de la courbe décrite par l'équation d'Attewell et Woodman (1982).
Formule utilisée : $s_{max} * G_{[m,i]}(x_f - x)$

Tassement tridimensionnel

Combiner l'expression des cuvettes transversale et longitudinale permet d'obtenir l'expression générale de la cuvette tridimensionnelle à une distance y de l'axe et une distance x du front du tunnel (Figure 1.12). Attewell et Woodman (1982) l'écrivent comme suit :

$$s(x,y) = s_{max} \cdot \exp\left[-\frac{1}{2}\left(\frac{y}{i_y}\right)^2\right] \cdot G_{[m,i_x]}(x_f - x) \tag{1.9}$$

Il convient de noter cependant que la formule originale considère une valeur unique de i ($i_x = i_y = i$). L'équation proposée ici se veut donc plus générale. La pertinence de cette proposition, qui découle de nos propres observations, sera discutée ultérieurement.

Progression du tassement avec l'avancement

Un changement de référentiel permet d'exprimer de la même façon le tassement observé en un point fixe de l'espace (x, y) en fonction de l'avancement de la position du front x_f. Cette description introduit une dépendance vis-à-vis du temps. Par simplicité, dans la suite on appellera cette description « équation de progression du tassement ». L'intérêt est de décrire ainsi les mesures en surface obtenues pour une cible visée de façon régulière dans le temps. La Figure 1.13 illustre les conventions d'écriture dans ce cas.

$$s(x_f) = s_{max} \cdot \exp\left[-\frac{1}{2}\left(\frac{y}{i_y}\right)^2\right] \cdot G_{[m,i_x]}(x_f - x) \tag{1.10}$$

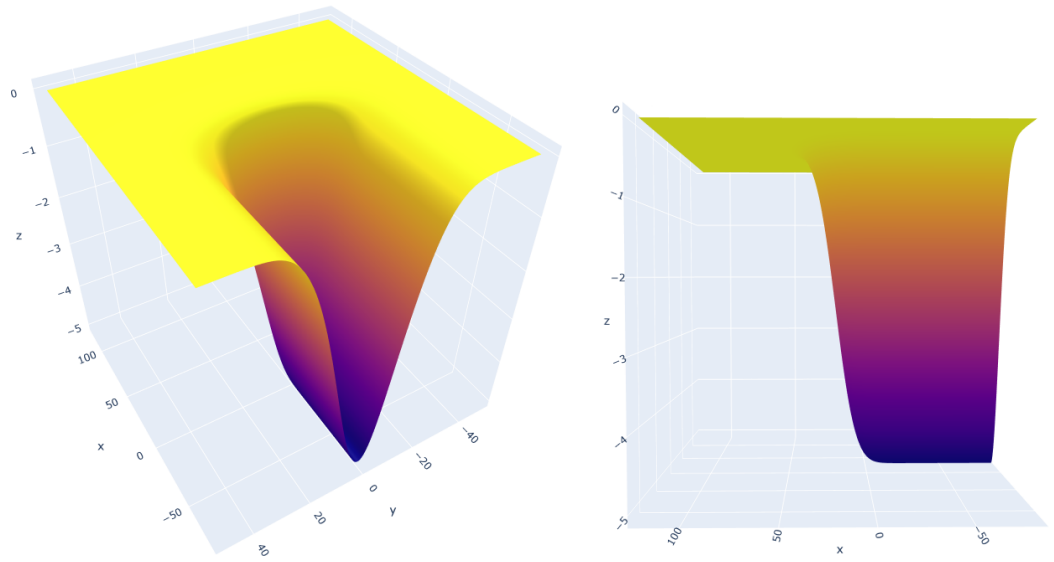


Figure 1.12. Tassement observé en tout point (x, y) de la surface du massif pour un creusement au tunnelier avec le front à la position $x_f = 20$

1.3 Méthodes de prévision des tassements

Les déformations induites en surface par le creusement des tunnels dépendent de nombreux facteurs liés à la géométrie du tunnel, aux conditions du sol, etc. selon une relation non-triviale, ce qui rend complexe la prédiction des tassements. Cette partie présente les avantages et inconvénients des méthodes traditionnelles de calcul des tassements : empirique, physique (modèle centrifugé), analytique et numérique. Par la suite, l'apport des méthodes basées sur l'Intelligence Artificielle est introduit.

1.3.1 Méthodes traditionnelles

Méthodes empiriques

Les méthodes de calcul empirique reposent sur l'observation et l'expérience plutôt que sur des principes théoriques. Les formules empiriques sont donc des expressions approximatives ajustées à partir de mesures observées. Les méthodes empiriques les plus classiques pour prédire les tassements sont celles élaborées par Peck (1969) et Attewell et Woodman (1982) et présentées précédemment dans le § 1.2.2. Selon ces formulations, la distribution du tassement en surface au-dessus du tunnel est représentée par une courbe gaussienne dans la direction transversale et par une courbe gaussienne cumulée dans la direction longitudinale (Figure 1.8). Cette approche sert de référence pour la validation d'autres approches. La formule de Peck en particulier est considérée comme fondamentale pour les méthodes empiriques et est ainsi devenue la base d'autres recherches (Attewell et Woodman, 1982 ; Cording et Hansmire, 1975 ; O'Reilly et New, 1982). Neaupane et Adhikari (2006) et Zhou et al. (2017) résument dans leurs travaux de nombreuses

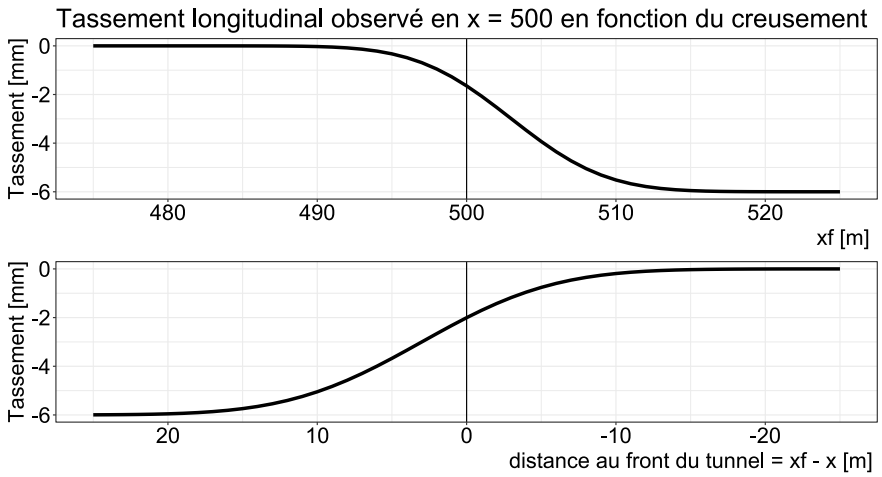


Figure 1.13. Illustration de l'équation de progression du tassement en fonction du creusement. Légende : x, x_f

relations empiriques ou semi-empiriques développées au fil des ans pour la prédiction des tassements.

Modélisation physique

Les méthodes basées sur la modélisation physique sont des simulations sous une gravité normale ou bien en centrifugeuse (Bel, 2018). Elles permettent de modéliser expérimentalement le creusement de tunnels à échelle réduite pour étudier le mouvement du sol induit par le creusement dans différents types de sols ainsi que les facteurs ayant la plus grande influence sur le comportement sol-tunnel. Ces modélisations physiques simplifient le contexte géotechnique et les conditions de creusement pour mieux comprendre les mécanismes en jeu. Pour le cas d'un creusement pressurisé, différents types de modèles physiques existent avec des niveaux de complexité variable (Berthoz, 2012 ; Berthoz et al., 2020). Les données produites peuvent contribuer au calage de modèles numériques.

Modèles analytiques

Au cours des dernières décennies, plusieurs solutions analytiques pour calculer les tassements induits par le creusement des tunnels ont été développées. Verruijt et Booker (1998) ont proposé une solution analytique pour calculer le tassement dans des demi-espaces élastiques homogènes, solution qui avait été proposée à l'origine par Sagaseta (1987) pour le cas de perte de volume en souterrain (ground loss). D'autres solutions élastiques sont développées par Bobet (2001) et Loganathan et Poulos (1998) pour un tunnel peu profond dans un sol saturé à partir de la solution proposé par Einstein et Schwartz (1979) pour un tunnel profond dans un sol sec.

Modélisation numérique

Avec l'augmentation de la puissance de calcul et des capacités de mémoire des ordinateurs au cours des dernières décennies, la modélisation numérique des ouvrages géotechniques, notamment par la méthode des éléments finis (EF) ou des différences finies, a été largement développée et utilisée (Bourgeois et al., 2018). Les modèles élaborés ont pour objectif de contribuer à l'analyse de conditions géométriques complexes et à la compréhension des différentes sollicitations auxquelles seront soumis les ouvrages.

Comparée à d'autres méthodes de prédiction des tassements, la modélisation numérique permet d'évaluer les tassements dans des situations complexes sans recourir à des simplifications. En effet, la simulation du creusement de tunnels à l'aide des modèles aux EF bidimensionnels ou tridimensionnels est capable de prendre en compte les caractéristiques de la construction (processus de construction, facteurs géométriques du tunnel et paramètres opérationnels, phasage d'excavation), les types de soutènement ainsi que les conditions du sol (géométrie, contraintes initiales, comportement non linéaire du sol, hydrologie etc.) avec des modèles de comportement du sol sophistiqués. Les méthodes des EF 2D ou EF 3D sont ainsi devenues des méthodes numériques populaires pour étudier les déplacements induits par un tunnelier.

Modèle EF 2D : L'analyse 2D en coupe transversale du tunnel et en déformations planes est la plus souvent utilisée. Différentes approches existent : convergence-confinement (CV-CF), contraction volumique et pression d'injection modifiée. Pour valider les modèles EF 2D, les résultats sont souvent comparés aux mesures des terrains ou bien à des modèles EF 3D (Dias et Kastner, 2013 ; Do et Dias, 2017 ; Gilleron, 2017 ; Likitlersuang et al., 2014).

Compte-tenu de leur facilité et rapidité de mise en œuvre, ces approches 2D sont largement utilisées dans la pratique courante de dimensionnement des tunnels (Gilleron et Bourgeois, 2018 ; Karakus, 2007 ; Moller et Vermeer, 2008). La méthode convergence–confinement se distingue par le fait de prendre en compte l'aspect tridimensionnel du problème du creusement d'un tunnel par une simplification basée sur le concept du taux de déconfinement λ (AFTES, 2002 ; Panet et Guenot, 1982 ; Panet, 1995 ; Panet et Sulem, 2021). Dans le cas d'un creusement au tunnelier avec une pression de stabilisation au front, il est d'usage de corriger le taux de déconfinement classiquement calculé par la pression de confinement du tunnelier rapportée à la contrainte initiale in situ (Aristaghes et Autuori, 2001).

Modèle EF 3D : D'après Janin (2017) et Berthoz et al. (2020), la modélisation tridimensionnelle permet d'étudier le problème dans toute sa complexité et se rapprocher au mieux de la réalité. Il est ainsi possible de prendre en compte notamment l'hétérogénéité du massif, la géométrie réelle de l'ouvrage, le phasage des travaux, les différentes pressions exercées autour du tunnelier, l'interaction avec les structures existantes. En particulier, l'approche 3D est non seulement capable de simuler la tridimensionnalité du champ des déformations généré autour du tunnel, mais éga-

lement de modéliser correctement les évolutions des contraintes autour du front de taille, les déplacements dans le massif et le chargement du soutènement et du revêtement (El Jirari, 2021 ; Kasper et Meschke, 2004 ; Migliazza et al., 2009 ; Mroueh et Shahrour, 2008).

1.3.2 Vers des méthodes innovantes

Limitations des méthodes traditionnelles

Les méthodes traditionnelles sont très utilisées mais ont des limitations.

Méthodes empiriques : les formules déduites de ces méthodes requièrent des paramètres qui sont difficiles à déterminer. Certains paramètres sont donnés par des abaques en fonction de la nature des sols rencontrés. Toutefois, les valeurs choisies restent approximatives et basées sur des hypothèses simplificatrices. De plus, ces modèles ne sont pas adaptés aux problèmes complexes. Les formulations mettent généralement en corrélation les mouvements du sol avec une ou deux variables uniquement. Cela peut être dû à des restrictions inhérentes à l'expression mathématique. C'est le cas par exemple des terrains multi-couches qui constituent un problème complexe avec de nombreux paramètres. En conséquence, les modèles empiriques ne sont pas adaptés à la complexité des conditions géologiques et des méthodes modernes de creusement. C'est particulièrement vrai pour l'excavation au tunnelier et ses nombreux paramètres de pilotage, étroitement reliés aux tassements observés en surface : pression au front, vitesse d'avancement, quantité de mortier injecté à l'arrière du front, etc.

Modélisation physique : Malgré ses qualités, la centrifugation des modèles n'est pas sans inconvénients. L'étude de Taylor (1995) met en évidence certaines limitations associées aux modèles réduits, telles que les effets d'échelle liés à la taille des grains utilisés dans la réalisation de modèles réduits. De plus, l'homogénéité des contraintes dans les massifs reconstitués est imparfaite en raison du gradient de force centrifuge présent dans la largeur du massif. Par ailleurs, il est important de choisir les matériaux de modélisation avec soin afin de garantir le respect des lois de similitude qui sont essentielles pour transposer les résultats obtenus à l'échelle du modèle réduit au chantier (Bel, 2018). Finalement, des difficultés d'ordre pratique existent telles que la difficulté de reconstituer des massifs représentatifs de géologies complexes, le coût et le temps nécessaire à la réalisation de l'essai ainsi que la disponibilité des centrifugeuses dans le monde.

Méthodes analytiques : Ces méthodes sont relativement simples dans leur mise en œuvre mais limitées dans la description de la relation non linéaire complexe entre les tassements et les différents paramètres qui influent sur les déformations. En effet, les méthodes analytiques ont été développées sur la base des équations fondamentales de la théorie élastique. Ces solutions sont donc en toute théorie limitées au cas de

l'excavation des tunnels dans des sols élastiques homogènes. Dans des conditions géologiques complexes, leurs prédictions ne sont en général pas applicables, ou nécessitent des simplifications en amont ce qui en réduit l'intérêt.

Modélisation numérique : La modélisation numérique du creusement des tunnels est limitée par plusieurs aspects. Tout d'abord, le choix des lois de comportement appropriées et des paramètres de sol adaptés n'est pas évident. En effet, le choix de la loi de comportement demande beaucoup de pratique et d'expérience et nécessite d'avoir des données adaptées et de qualité pour décrire avec confiance le comportement des matériaux. Comme les informations disponibles sur les propriétés du sol sont rares dans de nombreux cas, le modèle de comportement du sol modélisé peut souvent diverger de son comportement réel. Il s'en suit donc dans ces cas là des différences entre les prévisions réalisées a priori et les constats et mesures réalisés a posteriori au moment des travaux.

Ensuite, la simulation des différentes étapes du processus d'avancement d'un tunnelier est complexe. Le processus de construction de tunnel est toujours difficile à simuler. La mise en place d'un modèle réaliste qui serait capable de simuler la séquence d'excavation, la procédure d'installation du revêtement, l'injection et le renforcement, les effets d'échelle et de temps sont des difficultés souvent mentionnées. Il faut garder à l'esprit que même la simulation la plus sophistiquée restera toujours une simplification de la réalité. Par ailleurs, la précision des calculs dépend fortement de la sélection d'un maillage du modèle éléments finis approprié en termes de type et de taille des éléments (géométrie, nombre de noeuds).

D'autre part, pour la méthode convergence-confinement utilisée très souvent dans les modèles EF 2D, il s'avère que les hypothèses de base de la méthode sont souvent non respectées. En effet, la méthode CV-CF est couramment utilisée dans des situations de tunnel peu profond, de massif hétérogène et anisotrope, ou d'excavation phasée... A cela s'ajoute que le choix du taux de déconfinement λ de la méthode CV-CF dépend fortement du sol et de la loi de comportement utilisée (Gilleron et al., 2017 ; Moller et Vermeer, 2008). Le facteur λ doit être correctement calibré, par exemple par un calcul 3D ou par des retours d'expérience. Concernant les modèles EF 3D, des analyses tridimensionnelles abouties avec des modèles constitutifs de sol non linéaires avancés nécessitent des temps et puissances de calcul considérables, ainsi que des données de sols suffisamment nombreuses et précises.

Au final, les résultats des modèles numériques diffèrent souvent des mesures sur le terrain et des études de cas. Il est primordial de concevoir les ouvrages en restant conscient des incertitudes qui subsistent inévitablement, puis de construire en adaptant les techniques à la réalité des chantiers (Guilloux, 2016). Cette idée de mettre en pratique la méthode observationnelle le plus souvent possible pour prévenir les risques avec des rétro-calages réguliers se heurte à la réalité de la durée requise pour réaliser ces rétro-calages en phase chantier

Méthodes innovantes basées sur l'IA

Pour s'affranchir des limites des méthodes traditionnelles citées ci-avant, des méthodes basées sur l'Intelligence Artificielle (IA), plus précisément sur l'apprentissage automatique ou Machine Learning (ML), peuvent être utilisées. L'intérêt de ces méthodes est la possibilité de tenir compte des incertitudes et de la variabilité des propriétés du terrain, ce qui permet d'espérer plus de fiabilité. De plus, ces outils systématiques permettent de traiter toute la complexité des problèmes, en s'affranchissant du problème de modélisation des comportements physiques sous-jacents. Enfin, grâce à l'exploitation des mesures in-situ, ces outils permettent une réévaluation quasi-instantanée des variables utilisées.

Ces techniques sont regroupées sous le terme « Soft Computing » qui se traduit par « calcul souple » pour renvoyer à leur nature d'outil informatique puissant et polyvalent capable de traiter des informations incertaines ou imprécises et de s'adapter. En effet, contrairement aux techniques de « hard computing » traditionnelles, qui reposent sur des règles déterministes et des formules mathématiques précises, les techniques de Soft Computing utilisent la logique floue, le raisonnement probabiliste et d'autres techniques pour gérer l'ambiguïté, la variabilité et l'incertitude des données. L'apprentissage automatique a donc l'avantage de résoudre des problèmes non linéaires présentant un grand nombre de dimensions. En d'autres termes, un algorithme de ML est capable de trouver les corrélations non-linéaires entre de nombreux paramètres d'entrée dans le but de prédire un (ou des) paramètre(s) de sortie. Dans le cas de la prédiction des tassements induits par le creusement des tunnels, le paramètre de sortie est le tassement et les paramètres d'entrée sont tous les paramètres qui selon l'ingénieur ont une certaine influence sur le tassement, notamment les paramètres liés à la méthode d'excavation. Nous explorerons cette voie dans la suite de cette thèse.

Ces méthodes ont néanmoins également leurs limitations. Elles ont évidemment besoin d'une quantité de données importante pour entraîner des modèles qui assurent des niveaux de fiabilité satisfaisants. Ce sont par ailleurs intrinsèquement des boîtes noires pour lesquelles il n'est pas possible d'expliquer la façon dans les rouages internes se sont ajustés pour donner tel ou tel résultat.

Une introduction à la méthode de ML ainsi que l'état de l'art de la prédiction des tassements à l'aide d'algorithmes d'apprentissage automatique seront abordés dans le Chapitre 2 et le Chapitre 3 respectivement.

Conclusion

Ce chapitre était dédié à 3 sections distinctes visant à décrire le tunnelier à pression de terre, les déformations en surface causées par le creusement des tunnels et les méthodes traditionnelles de calcul du tassement.

Dans un premier temps, une description de l'historique des tunnels depuis l'antiquité

jusqu'à l'utilisation des méthodes mécanisées a été présentée. Ensuite, une explication détaillée des tunneliers à pression de terre a été fournie en prenant comme exemple les tunneliers utilisés pour l'excavation d'un tronçon de la ligne 15 du projet du Grand Paris Express.

Dans la deuxième section, les sources de déformation en surface causées par le creusement des tunnels ont été décrites, avec rappel et résumé des équations de tassement transversal et longitudinal. L'hypothèse d'équivalence espace-temps a été mise en valeur, permettant ainsi de présenter clairement l'équation de progression du tassement. Cette équation représente le tassement observé en un point de l'espace en fonction de l'avancement du creusement du tunnel.

Enfin, dans la troisième section, les méthodes traditionnelles de calcul du tassement ont été décrites, tout en soulignant leurs limitations. L'intérêt de l'utilisation des méthodes d'Intelligence Artificielle pour la prévision des tassements a ainsi été mis en évidence.

En résumé, ce chapitre constitue une étape importante dans notre compréhension des tunnels, des déformations en surface et des méthodes de calcul du tassement, nous permettant ainsi d'explorer de nouvelles pistes de recherche pour réduire les risques relatifs au creusement des tunnels au tunnelier.

Le chapitre suivant sera consacré à l'explication des notions fondamentales de l'Intelligence Artificielle, à destination des ingénieurs-utilisateurs, dans le but de permettre une prise en main aisée de ces notions dans la suite de cette thèse.

Dans le monde de l'Intelligence Artificielle, de l'apprentissage automatique et de la science des données

2

Introduction

L'intelligence artificielle IA est l'un des domaines les plus en vogue de nos jours, avec une grande variété d'applications potentielles. Depuis ses débuts dans les années 1950, l'IA a connu une évolution rapide, stimulée par des avancées technologiques, des données plus abondantes et des algorithmes plus sophistiqués. Cette évolution a conduit à des percées majeures dans la reconnaissance d'images, le traitement du langage naturel, la prédiction de séries temporelles et bien d'autres domaines. L'IA est aujourd'hui un terme qui reste nébuleux et fantasmatique pour beaucoup de gens, notamment ceux qui n'ont pas une formation en informatique. Pourtant, l'IA est devenue une technologie clé dans de nombreux domaines, y compris en ingénierie géotechnique. En effet, les problématiques rencontrées en géotechnique, telles que la prédiction des mouvements de terrain ou la stabilisation des excavations, sont souvent complexes et difficiles à résoudre à l'aide de méthodes traditionnelles. L'IA offre une nouvelle approche pour aborder ces problèmes en utilisant des techniques d'apprentissage automatique pour extraire des modèles à partir de données.

Ce chapitre vise à fournir une introduction synthétique et opérationnelle à l'IA, en commençant par son historique et en passant en revue les définitions clés telles que l'apprentissage automatique et le Big Data. L'importance des données et des bases de données, ainsi que l'empreinte carbone associée à la collecte et au stockage de données massives sont également abordées. Ensuite, on décrit les différentes catégories d'apprentissage automatique telles que l'apprentissage supervisé ou non-supervisé, la régression et la classification, ainsi que les concepts clés tels que le sur-apprentissage et la validation croisée. Enfin, des algorithmes d'apprentissage automatique tels que les réseaux de neurones artificiels sont décrits. Par ce chapitre, nous espérons démystifier le monde de l'IA pour les ingénieurs qui souhaitent utiliser ces méthodes dans leur pratique professionnelle.

2.1 Intelligence Artificielle et Données

L'Intelligence Artificielle, connue sous son acronyme IA, est l'exploitation des machines pour imiter les capacités de résolution de problèmes et de prise de décision du cerveau humain. L'apprentissage automatique ML (Machine Learning) est un domaine d'application de l'IA.

Avant d'entrer au cœur de ce travail de thèse, ce chapitre permet de résumer les éléments de contexte et la sémantique associée aux concepts d'Intelligence Artificielle et d'apprentissage automatique. Il sera ensuite discuté de l'importance des données pour l'application de ces méthodes. Seront enfin abordées les problématiques subséquentes à la gestion de la donnée ce qui conduira à expliquer la plus-value apportée par l'utilisation des bases de données structurées.

2.1.1 Éléments de cadrage

Rappels historiques

Les bases de l'IA sont posées dès 1950 par Alan Turing dans son article « Computing Machinery and Intelligence » publié dans le journal britannique *Mind* (Turing, 1950). Cette publication est reconnue pour le fameux "test de Turing" où l'auteur définit un test permettant de déterminer si une machine est intelligente ou pas. En résumé, selon le test de Turing, une machine est considérée comme intelligente si elle est capable de se faire passer pour un humain lors d'une conversation. Cette formulation du concept d'intelligence machine n'est présentée que comme un simple postulat naïf, une simplification arbitraire qui a vocation à provoquer les confrères les plus sceptiques de l'époque de Turing (un « jeu de l'imitation », selon les propres termes de Turing) (Castelfranchi, 2013).

En 1956, la terminologie « Intelligence Artificielle » est proposée pour la première fois par John McCarthy lors d'une conférence du *Dartmouth College* aux Etats-Unis (Moor, 2006). Les scientifiques étaient alors persuadés que l'avènement de cerveaux électroniques égalant l'Homme était imminent. Vingt ans après, on a dû admettre que les ordinateurs des années 70 restaient primitifs et incapables de réaliser ces attentes utopiques. Cette période est connue sous le nom d'« hivers de l'IA ». Ce n'est qu'en 1995 que le hardware atteint enfin un seuil en matière de performances. Les investissements dans le domaine de l'IA reprennent. En 1997, l'ordinateur Deep Blue développé par IBM bat le champion du monde d'échecs. L'IA a pris une nouvelle dimension à partir de 2010 et cela s'explique par deux facteurs : l'accès à des volumes massifs de données et la découverte de la très grande efficacité des processeurs de cartes graphiques des ordinateurs pour accélérer le calcul des algorithmes d'apprentissage pour un coût financier restreint (capables de plus de mille milliards d'opérations par seconde). De nombreux exploits de l'IA ont eu lieu dans les années qui ont suivi. La Figure 2.1 récapitule les principales dates et événements discutés dans cet état de l'art.

Types d'IA et Réglementation

On distingue 2 types d'IA : l'IA faible et l'IA forte. Une IA faible est une IA entraînée pour une tâche ciblée. C'est la grande majorité des IA qui nous entourent aujourd'hui (Figure 2.3). Quant à l'IA forte, elle reste une forme théorique de l'IA dans laquelle une

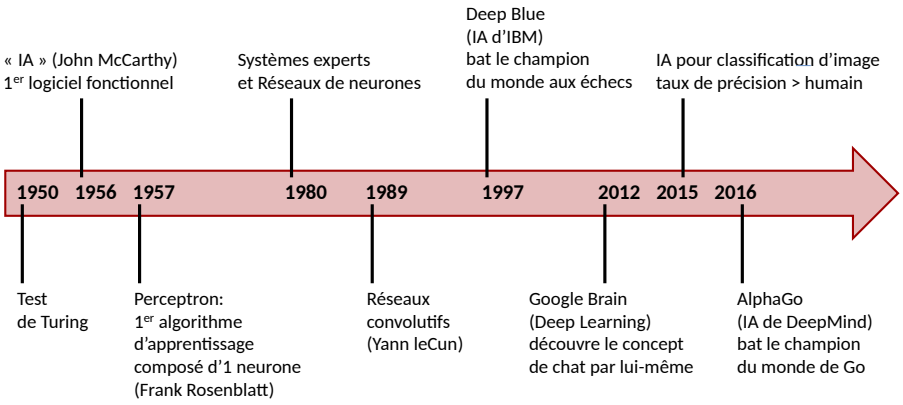


Figure 2.1. Dates clés dans l'histoire de l'IA

machine aurait une intelligence et des capacités égales ou supérieures à celles du cerveau humain. Elle aurait conscience d'elle-même, serait capable de résoudre des problèmes pour lesquels elle n'aurait pas été entraînée spécifiquement et serait capable d'apprendre et de planifier l'avenir. De nombreuses questions éthiques et philosophiques se posent alors et font l'objet de plusieurs ouvrages comme par exemple le livre "La guerre des Intelligences" (Alexandre, 2017).

L'Union Européenne travaille à cadrer par la réglementation l'usage de l'IA dans les secteurs les plus sensibles des services et de l'industrie (ingénierie, éducation, santé, transport, etc.). Il s'agit de tirer le meilleurs parti de la valeur ajoutée lié à ce usage et de circonscrire les risques potentiels à travers des recommandations et guides de bonnes pratiques pour permettre l'émergence d'IA de confiance et d'excellence (Quantmetry, 2021). L'historique de cette réglementation est donnée dans la Figure 2.2.

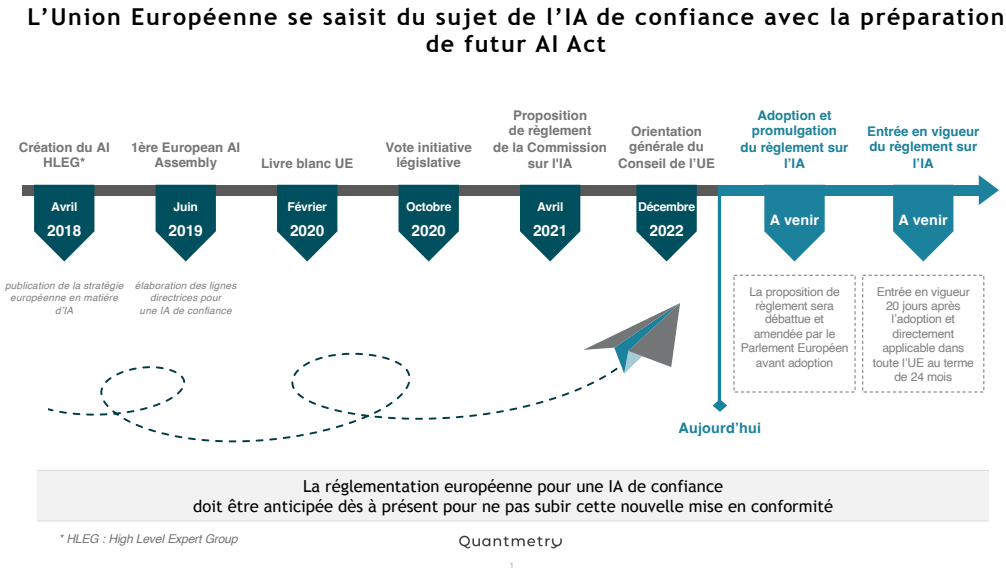


Figure 2.2. Les étapes de réglementations de l'UE sur l'IA de confiance (Quantmetry, 2023)

Intelligence Artificielle, Machine Learning et Big Data : Définitions

Intelligence Artificielle (Artificial Intelligence) : Selon la CNIL (Commission Nationale de l'Informatique et des Libertés) et le Parlement européen (CNIL, 2022), l'Intelligence Artificielle est un domaine scientifique qui regroupe tout outil utilisé par une machine afin de « reproduire des comportements liés aux humains, tels que le raisonnement, la planification et la créativité ». Tout système mettant en œuvre des mécanismes proches de celui d'un raisonnement humain pourrait ainsi être qualifié d'Intelligence Artificielle. Cette définition implique que tout système qui n'est pas dotée de capacité d'apprentissage à partir de données ou de capacité de prise de décision n'est pas une Intelligence Artificielle. On pense notamment dans notre secteur aux algorithmes d'automatisation, aux tableurs, aux SIG (Systèmes d'Information Géographiques), au BIM (Building Information Modelling), aux solutions de stockage et de distribution instantanée de données (ex : outils de suivi de chantier en direct), à la réalité virtuelle, à l'impression 3D, aux jumeaux numériques, etc. Ces méthodes peuvent plutôt être classées dans la catégorie de « transformation digitale » (Digital Transformation) ou du « numérique ».

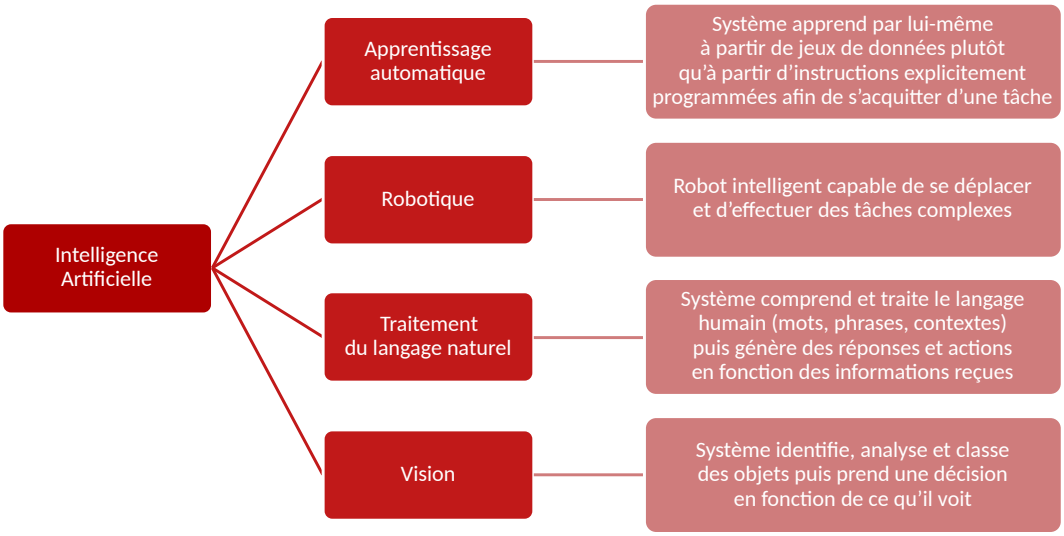


Figure 2.3. Exemples de domaines de l'Intelligence Artificielle

Apprentissage Automatique (Machine Learning) : Parmi les domaines d'application de l'IA (Figure 2.3), on distingue la branche de l'apprentissage automatique ou Machine Learning (ML).

Dans son article de 1950 (Turing, 1950), Turing prédit déjà l'apprentissage des machines (« Learning Machines » selon ses termes) comme « la simulation du cerveau d'un enfant plutôt que celui d'un adulte ». La machine aurait ainsi besoin d'un programme (le cerveau) et d'apprentissage (éducation) pour déduire des relations entre différents paramètres. Il souligne aussi l'importance d'expérimenter en enseignant à plusieurs machines afin de trouver celle qui réussit à résoudre le problème d'une façon optimale.

En 1959, Arthur Samuel définit le ML comme « le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre [à partir de données] sans être explicitement programmés ». En effet, le ML n'est fondamentalement rien de plus que la capacité d'un programme à évaluer statistiquement un résultat à partir d'un jeu de données, et à réduire l'écart entre ce résultat et la valeur vraie par un jeu d'apprentissage itératif. Cela signifie que, au lieu de programmer manuellement des règles et des algorithmes pour effectuer une tâche précise, on peut utiliser le ML pour créer des programmes qui peuvent s'acquitter de cette tâche en s'ajustant automatiquement à partir de jeux de données. On déduit que l'apprentissage de la machine nécessite beaucoup de données pour trouver les des relations précises entre les paramètres influençant la tâche étudié. Ce n'est donc pas par hasard que l'essor du ML soit lié à celui du Big Data.

Données massives (Big Data) : On parle depuis quelques années du phénomène de Big Data. Ce dernier se traduit souvent par « données massives » préférablement au terme « mégadonnées » qui fait référence à une quantité de l'ordre d'un million (en grec, méga désigne 10^6). Le Big Data désigne donc des ensembles de données devenus si volumineux qu'ils dépassent les capacités humaines d'analyse et même celles des outils informatiques classiques. L'exploitation de ces ensembles de données a nécessité le développement de technologies spécifiquement adaptées. Pour fixer les idées, on peut parler de Big Data lorsque le volume de données analysé dépasse la capacité d'un simple ordinateur personnel. Le Big Data est défini par cinq notions qu'on appelle les « 5V » (Bourany, 2019) :

1. Volume : avec le développement des nouvelles technologies comme les IoT (Internet of Things : appareils physiques qui reçoivent et transfèrent des données sur des réseaux sans fil, avec une intervention humaine limitée), la production de données numériques a été de plus en plus massive : textes, photos, vidéos, etc.
2. Vélocité : besoin d'analyse des données en temps réel et de prise de décision en une fraction de seconde
3. Variété : les données d'intérêt sont hétérogènes par nature, avec un format structuré ou non : vidéos, géolocalisation, échanges vocaux, messages sur les réseaux sociaux, etc.
4. Véracité : l'incertitude associée aux données doit être quantifiée en termes de précision, de validité, de cohérence, de fiabilité et de qualité.
5. Valeur : c'est un des points les plus importants à notre avis mais rarement mentionné. L'utilisation des technologies de stockage et d'analyse des Big Data n'a de sens que si elle apporte de la valeur ajoutée : exploiter les données, c'est avant tout répondre à des objectifs métiers.

Par conséquent, la génération et le stockage des Big Data a favorisé l'émergence de méthodes statistiques, qui traitent de gigantesques volumes de données pour en

tirer du sens et en créer de la valeur. C'est donc un tournant majeur de la science des données (Data Science), du ML et de l'IA.

2.1.2 Données : le cœur du sujet

La qualité des programmes développés à l'aide de méthode de Machine Learning dépend complètement de la qualité et de la structuration des données sources. C'est ce qui a inspiré au mathématicien britannique Clive Humby la comparaison suivante : « data is the new oil » (la donnée est le nouveau pétrole) (ANA Senior marketer's summit, Kellogg School, 2006). Cette comparaison fait par ailleurs échos à l'idée que la valeur de la donnée n'est pas intrinsèque (au même titre que la nature nous fourni le pétrole gratuitement), et que c'est les travaux d'extraction et de traitement de celle-ci qui lui donnent sa valeur.

Source et traitement des données

Dans le domaine de la construction, la donnée provient de nombreuses sources comme les plans, les notes de calcul, les capteurs, l'instrumentation du sol, les données d'essais in-situ ou en laboratoire, les logs des sondages, les stations météorologiques, les équipements GPS, l'instrumentation embarquées sur les engins, etc. Ces moyens de mesure permettent d'acquérir un grand nombre de données au fil du temps. Il existe également des « banques de données » comme par exemple la base de données SONGE du Projet du Grand Paris Express § II construite à partir de nombreux sondages effectués lors de la campagne de reconnaissance des sols.

Toutefois, les données provenant des chantiers ont toujours besoin d'un traitement préalable pour écarter les erreurs de mesures et valeurs aberrantes de toutes origines. Ces dernières proviennent notamment des conditions météorologiques, du niveau d'intervention humaine et des erreurs de mesures des instrumentations. Cela induit la génération de données qui contiennent des erreurs telles que des fautes d'orthographe ou de ponctuation, des données incorrectes, des données périmées, des données dupliquées et des données aberrantes (outliers). C'est ce qu'on désigne sous le terme générique de données « sales » (dirty data) : des données inexactes, incomplètes ou incohérentes qui nécessitent un nettoyage avant leur utilisation pour des analyses postérieures. Le nettoyage des données améliore la qualité du jeu de données (dataset) ce qui renforce la confiance dans l'ensemble des données et la fiabilité des résultats d'analyse de données (Chu et al., 2016 ; Klein et Lehner, 2009 ; Krishnan et al., 2016). Cependant, le traitement des données est un travail fastidieux et chronophage. En effet, des études de IBM (2016) et CrowdFlower (2016) concluent que la collecte, le nettoyage et l'organisation des données constituent 80% du temps de travail d'un data scientist.

Pour faire face au manque de données réelles, on utilise fréquemment la génération de données à partir de modèles théoriques, empiriques ou numériques. Cela permet de simuler de nombreux cas d'étude et différents scénarios avec différents paramètres afin d'obtenir les résultats associés. La génération de données permet notamment de combler

un manque de données d'une catégorie spécifique. A titre d'exemple on peut citer des travaux portant sur la collecte de données de radiers de bâtiments, les charges qui leur sont appliquées, la nature des sols d'assise et les tassements mesurés. Les données de chantier de cet exemple proviennent uniquement de radiers bâtis sur des sols parisiens. Dans ce cas, on a utilisé la génération de données par des méthodes numériques pour compléter la base de données avec des cas de radiers dans des sols de différentes natures.

Stockage et base de données

Après l'obtention des données, il est important de réfléchir à leur stockage, ce qui permet un nettoyage et un traitement facilités. Dans le domaine de la construction, il est souvent d'usage d'utiliser des feuilles de calcul type csv ou Excel. Il existe également des formats intermédiaires, structurées et dont la nomenclature est documentée voire standardisée, comme le format AGS (Association of Geotechnical and Geoenvironmental Specialists) (AGS, 2022). Ce format décrit la sémantique et l'organisation des fichiers à adopter pour que celui-ci puisse être lu par des logiciels le supportant, ou n'importe quel utilisateur ayant créé des outils adaptés à son exploitation. Format « multitable » et hiérarchisé, il peut s'interfacer de façon naturelle avec des BD hiérarchisées. Dans le domaine de la géologie, de nombreux autres formats de ce type existent (GeoSciML, RSQML, etc.). Ils s'appuient en général sur des structure de fichiers hiérarchisés comme XML ou JSON.

Concernant les feuilles de calcul, elles ont été conçues pour une utilisation individuelle ou pour un petit nombre d'utilisateurs qui n'ont pas besoin d'effectuer de nombreuses manipulations de données. La construction et l'utilisation de bases de données (BD) a donc énormément d'avantages par rapport à l'échange de fichier. Elles permettent à plusieurs utilisateurs d'accéder aux données en même temps, rapidement et en toute sécurité (Table 2.1), avec une gestion des droits et les rôles de chaque utilisateur intégrée. Les BD sont également conçues pour gérer de grandes quantités d'informations organisées.

Par définition, une base de données est une collection organisée d'informations structurées gérée par un Système de Gestion de Base de Données (SGBD ou Database Management Systems). Ce dernier assure la création, la modification, le stockage et la sécurité des données, ainsi que la gestion d'accès multiples. Parmi les SGBD les plus populaires on peut citer PostgreSQL, MySQL, Microsoft Access, Microsoft SQL Server et Oracle Database. PostgreSQL est considéré comme le SGBD le plus documenté disponible sous licence libre. L'ensemble que constituent les données et le SGBD, ainsi que les applications qui leur sont associées, est nommé BD.

Les BD permettent d'aller au-delà du stockage des données et des opérations basiques. Elles permettent d'analyser de grandes quantités de données provenant de différents systèmes en utilisant des outils de calcul et de business intelligence. Grâce à ces analyses, les entreprises peuvent utiliser les données collectées pour améliorer leur efficacité, optimiser leur prise de décision et devenir plus agiles et flexibles. Aujourd'hui, l'optimisation de l'accès aux données et de la vitesse de traitement est cruciale pour les entreprises, en

Table 2.1. Comparaison des feuilles de calcul et des bases de données

Caractéristique	Feuille de calcul (Excel, csv..)	Base de données
Structure des données	Les données sont stockées dans des cellules organisées en lignes et en colonnes	Les données sont stockées dans des entités (tables) avec des attributs (colonnes) et des tuples (lignes)
Manipulation des données	Les données peuvent être filtrées, triées et regroupées en utilisant des fonctions et des formules simples	Les données peuvent être filtrées, triées, regroupées, jointes et agrégées en utilisant des requêtes complexes
Partage des données	Les feuilles de calcul peuvent être partagées en tant que fichiers, mais peuvent être difficiles à gérer lorsqu'il y a beaucoup de collaborateurs	Les bases de données peuvent être partagées en ligne ou en réseau, ce qui les rend plus faciles à gérer lorsqu'il y a plusieurs utilisateurs
Sécurité des données	Les feuilles de calcul peuvent être protégées par mot de passe, mais cela peut ne pas être suffisant pour les données sensibles	Les bases de données peuvent être protégées par des mécanismes de sécurité tels que l'authentification, l'autorisation et le chiffrement des données
Analyse des données	Les feuilles de calcul peuvent effectuer des analyses simples en utilisant des graphiques et des tableaux croisés dynamiques, mais peuvent devenir complexes lorsqu'il y a beaucoup de données	Les bases de données peuvent effectuer des analyses complexes en utilisant des requêtes complexes et des outils d'analyse de données tels que SQL et Business Intelligence
Performance	Les feuilles de calcul peuvent devenir lentes lorsqu'elles contiennent beaucoup de données ou de formules complexes	Les bases de données sont conçues pour gérer de grandes quantités de données et peuvent être optimisées pour une meilleure performance

raison de la croissance exponentielle des quantités de données à gérer. Il est donc impératif d'avoir une plateforme capable de fournir les performances, l'évolutivité et l'agilité nécessaires pour soutenir ce levier de croissance de beaucoup d'entreprises.

Les différents types de BD comprennent les BD relationnelles, orientées objet, NoSQL et d'autres. Ce travail se concentre sur les BD relationnelles, qui sont le type de BD le plus couramment utilisé. Les BD relationnelles utilisent généralement le langage de programmation Structured Query Language (SQL) pour interroger, manipuler et définir les données et gérer les accès (Sumathi et Esakkirajan, 2007). Les données sont modélisées sous forme de lignes et de colonnes dans une série de tables, ce qui permet d'optimiser l'efficacité du traitement et de l'interrogation des données. Les BD permettent une gestion efficace des données, telles que la consultation, la modification, la mise à jour et l'organisation des données. Le lecteur intéressé pourra se reporter à Oracle (2022) pour de plus amples détails sur les bases de données.

Empreinte carbone de la donnée

L'empreinte carbone des données désigne l'impact environnemental associé à la génération, au traitement, au stockage et au transfert de données, mesuré en termes de consommation d'énergie et de production de gaz à effet de serre. La croissance exponentielle de la quantité de données produites et stockées dans le monde entier a un impact sur l'environnement, en raison de la consommation d'énergie nécessaire pour leur gestion et des matériaux nécessaire à la construction des serveurs, locaux, etc. Face à cette préoccupation environnementale, les entreprises et les gouvernements s'efforcent de minimiser l'empreinte carbone des données en adoptant des pratiques éco-efficaces, telles que l'utilisation de centres de données énergétiquement performants et le déploiement de technologies vertes (Bouley, 2010 ; Charret et al., 2022).
L'impact environnemental de la donnée dépend de divers facteurs, notamment la quantité de données générées et stockées, les modalités de traitement et de transfert, et les technologies utilisées. L'augmentation de la quantité de données peut avoir un impact environnemental négatif, principalement en raison de la consommation énergétique liée au traitement, au stockage et au transfert de ces données. Par exemple, le transfert de données sur de longues distances, comme celui via internet, peut consommer de l'énergie et renforcer l'empreinte carbone globale. Il est donc primordial de prendre en compte ces considérations environnementales lors de la conception et de l'utilisation des technologies de l'information et de la communication.

2.2 Apprentissage Automatique

L'utilisation de l'apprentissage automatique n'est possible qu'après compréhension des notions de bases. Cette section a pour objectif de définir rigoureusement les algorithmes d'apprentissage automatique, les modèles, les fonctions de pertes, la division des données, la validation croisée, les différentes catégories d'apprentissage automatique ainsi que les applications telles que la régression ou la détection d'anomalies.

2.2.1 Définitions générales

Modèle $\hat{f}$ et fonction de perte

Soient X et y des variables aléatoires liées par des lois de probabilité inconnues. L'objectif est de trouver la fonction de corrélation entre X et y, notée f, telle que :

$$y = f(X)$$

La fonction f étant inconnue, on cherche à l'approcher par un modèle, noté $\hat{f}$, de sorte que l'erreur entre la cible y et sa prédiction $\hat{y} = \hat{f}(X)$ soit minimale. Cette erreur est calculée à l'aide de la fonction de perte (loss function) (§ 2.3.2) qui retourne une certaine mesure de la différence entre y et $\hat{y}$. L'intérêt de trouver une bonne approximation de f

est la possibilité de calculer la variable cible y (target) à partir de n'importe quel jeu de variables explicatives $X = x_1, x_2, ..., x_n$ (features, encore nommées variables indépendantes, caractéristiques, attributs ou encore prédicteur). Ladite fonction de perte mesure alors l'erreur de prédiction du modèle $\hat{f}$ permettant ainsi d'estimer sa qualité. En résumé, on utilise un modèle $\hat{f}$ pour prédire y à partir de X ; l'erreur entre la prédiction et la réalité est mesurée par une fonction de perte.

Sous-apprentissage et Sur-apprentissage

Pour trouver le bon modèle $\hat{f}$, on prend un jeu de données de taille n, $A = (X_i, y_i)_{i \leq n}$, sur lequel on entraîne un algorithme d'apprentissage (machine learning algorithm, par exemple un réseau de neurones). L'algorithme s'entraîne sur le jeu de données A pour trouver une approximation $\hat{f}$ de f en minimisant la fonction de perte.

Le risque est que le modèle obtenu soit trop simple pour bien capturer les relations dans les données : c'est le sous-apprentissage (ou le sous-ajustement, underfitting). Le sous-apprentissage est généralement causé par un manque de complexité dans le modèle, ce qui peut être résolu en utilisant un modèle plus complexe ou en utilisant un algorithme d'apprentissage différent.

Un autre risque est le sur-apprentissage (ou le sur-ajustement, overfitting). C'est une situation où le modèle obtenu est trop adapté au jeu de données A et ne peut pas généraliser ses prédictions aux données qui ne font pas partie de A. En d'autres termes, le modèle apprend les détails et le bruit des données de A au lieu de capturer les tendances générales qui pourraient être utiles pour de nouvelles données.

Le sous-apprentissage et le sur-apprentissage se traduisent par des performances de prédiction médiocres pour de nouvelles données (Figure 2.4). Un bon modèle doit être capable de généralisation, c'est-à-dire de prédire de façon satisfaisante la sortie cible y pour tout jeu de données, y compris ceux qui n'ont pas servi à son entraînement.

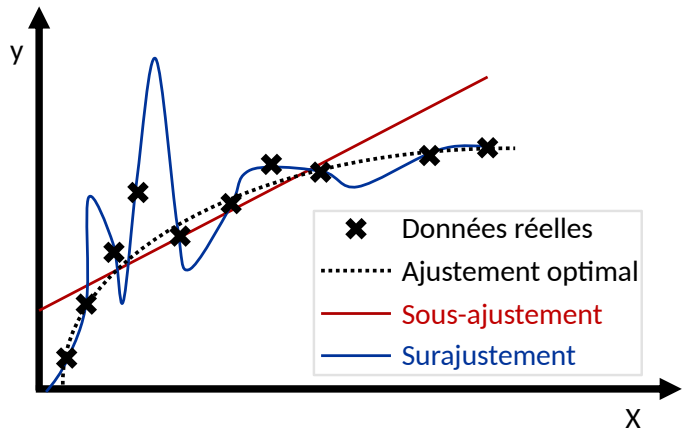


Figure 2.4. Sur-apprentissage (overfitting) et sous-apprentissage (underfitting) des données

Compromis biais-variance

La problématique du sous-apprentissage et du sur-apprentissage est liée aux concepts de biais et de variance. En effet, le biais correspond à l'erreur systématique d'un modèle qui découle de ses hypothèses simplificatrices. Un modèle avec un biais élevé sera trop simpliste pour décrire les relations complexes entre les variables d'entrée et de sortie, et sera donc susceptible de sous-apprendre.
Inversement, la variance correspond à la variabilité des prédictions d'un modèle en réponse à des variations dans les données d'entraînement. Un modèle avec une variance élevée sera très sensible aux fluctuations aléatoires des données d'entraînement, ce qui peut conduire à un sur-apprentissage.

Le compromis entre le biais et la variance est souvent appelé le compromis biais-variance (trade-off biais-variance). En général, un modèle avec un biais élevé aura une variance faible et vice versa. Le compromis consiste donc à trouver un modèle qui minimise à la fois le biais et la variance, de manière à obtenir une erreur de généralisation minimale (Figure 2.5).

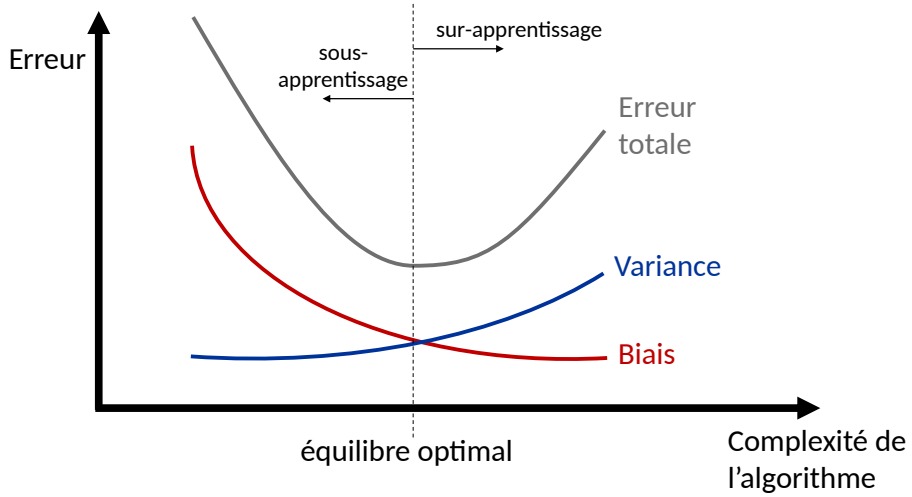


Figure 2.5. Évaluation de l'équilibre optimal par décomposition de l'erreur (inspirée de Nallaperumal (2021))

Division des données : ensembles d'apprentissage et de test

Pour vérifier la capacité de généralisation du modèle obtenu, on le teste sur un nouvel ensemble de données qu'il n'a jamais vu auparavant. Pour cela, on divise A en deux parties : une partie pour l'apprentissage (train) de l'algorithme d'apprentissage et une autre pour l'évaluation finale (test) du modèle obtenu. On obtient ainsi l'ensemble d'apprentissage (ou ensemble d'entraînement, training set) $A_{train} = (X_{train}, y_{train})$ et l'ensemble de test (ou ensemble d'évaluation, test set) $A_{test} = (X_{test}, y_{test})$. La démarche est alors la suivante :

1. On entraîne l'algorithme d'apprentissage pour trouver le meilleur modèle $\hat{f}$ correspondant au jeu de données d'apprentissage A_{train}

2. On utilise le modèle $\hat{f}$ ainsi trouvé pour prédire les données cibles de l'ensemble de test : y_{test}. On calcule alors l'erreur entre la cible y_{test} et sa prédiction $\hat{y}_{test}$: c'est l'erreur de généralisation (generalization error). Cela permet d'évaluer la performance de $\hat{f}$ sur des données jamais vues auparavant. Si l'erreur calculée est satisfaisante, on déduit que le modèle est de bonne qualité

Paramètres et Hyperparamètres

Un modèle $\hat{f}$ est en réalité un modèle complexe défini par un nombre de paramètres. Par exemple, un modèle linéaire simple est défini par deux paramètres qui sont la pente de la droite et l'ordonnée à l'origine. L'apprentissage consiste à trouver des valeurs optimales pour les paramètres du modèle $\hat{f}$, de sorte que $\hat{f}$ se généralise bien aux nouvelles données.

Un hyperparamètre est en revanche un paramètre de l'algorithme d'apprentissage lui-même, et non du modèle. C'est un ou plusieurs paramètres de configuration qui sont fixés avant l'apprentissage du modèle et qui ne sont pas appris directement à partir des données d'entraînement. Un exemple typique d'hyperparamètre est l'architecture d'un réseau de neurone (§ 2.3.1). La performance d'un modèle dépend fortement des hyperparamètres : il est donc indispensable de les optimiser.

Optimisation des hyperparamètres et régularisation

Il est primordial de distinguer les concepts d'optimisation et de régularisation des hyperparamètres en apprentissage automatique. L'optimisation des hyperparamètres (hyperparameters tuning) est le processus de recherche de la meilleure configuration des hyperparamètres d'un algorithme d'apprentissage automatique avant son entraînement. C'est un processus clé pour améliorer les performances des modèles d'apprentissage automatique en sélectionnant les valeurs optimales pour les hyperparamètres avant l'entraînement du modèle. Il est généralement effectuée à l'aide de techniques de recherche systématique telles que la recherche par grille (grid search) ou la recherche aléatoire (random search), qui consistent à explorer un ensemble prédéfini de valeurs possibles pour les hyperparamètres et à sélectionner ceux qui donnent les meilleurs résultats sur un ensemble de validation.

En revanche, la régularisation des hyperparamètres (regularization of hyperparameters) fait référence à l'utilisation de techniques de régularisation pour contrôler la complexité des modèles en ajustant les valeurs des hyperparamètres. La régularisation des hyperparamètres permet ainsi de trouver le bon compromis entre le biais et la variance pour éviter le sur-apprentissage d'un algorithme d'apprentissage automatique. Les techniques de régularisation courantes incluent la régularisation L1 (Lasso) et la régularisation L2 (Ridge) (Rakotomalala, 2023). Des exemples de régularisation et d'optimisation d'algorithmes à base d'arbres de décision sont présentés dans le § 6.2.3.

Ensemble de validation

L'optimisation des hyperparamètres ne doit pas être effectuée sur les données de test. Si c'était le cas, le modèle serait optimisé sur ce lot spécifique de données et aurait peu de chances d'être aussi performant sur de nouvelles données. Une solution courante à ce problème est la mise de côté d'un ensemble de données du lot d'apprentissage, appelé ensemble de validation (validation). Cette méthode est connue sous le nom de validation par exclusion (holdout validation) (données exclues pour la validation). La méthodologie de travail devient :

1. Division du lot de données A en deux ensembles distincts : un ensemble d'apprentissage et un ensemble de test. Il est d'usage de réserver 20% des données pour le test. Cependant, le choix de ce pourcentage doit être effectué avec précaution afin de garantir une quantité suffisante de données dans l'ensemble de test permettant d'évaluer de manière fiable la capacité de généralisation du modèle.
2. Exclusion d'une partie de l'ensemble d'apprentissage (par exemple 10%) pour s'en servir d'ensemble de validation.
3. Entraînement de plusieurs modèles avec différents hyperparamètres sur l'ensemble d'apprentissage réduit (c'est-à-dire l'ensemble d'apprentissage initial privé de l'ensemble de validation).
4. Sélection du modèle le plus performant sur l'ensemble de validation.
5. Entraînement du meilleur modèle sélectionné sur l'ensemble d'apprentissage complet (y compris l'ensemble de validation) : c'est le modèle final.
6. Évaluation de ce modèle final sur l'ensemble de test pour obtenir une estimation de l'erreur de généralisation.

Validation croisée

Pour éviter tout biais statistique lors de la division des données entre ensembles d'apprentissage et de validation, on utilise généralement la méthode de validation croisée (ou CV pour cross-validation). Il existe plusieurs types de CV : stratified k-fold CV, Leave-p-out CV, Monte Carlo CV, etc. La méthode la plus courante est la k-fold CV (Figure 2.6). Cette méthode consiste à partitionner l'ensemble des données A en k parties de tailles égales. Chaque partition est appelée un pli (ou fold) ce qui éclaire le nom de la méthode qui renvoie au fait qu'il y a k plis. Cette méthode présente également l'avantage de permettre l'entraînement et la validation du modèle sur des ensembles de données différents, ce qui peut réduire le risque de sur-apprentissage et fournir une évaluation plus précise de ses performances. La procédure générale est la suivante :

1. Division du lot de données A en deux : un ensemble d'apprentissage et un ensemble de test.
2. Mélange de l'ensemble des données d'apprentissage de façon aléatoire (cette étape n'est pas obligatoire mais recommandée pour réduire les aléas statistiques).

3. Division de l'ensemble des données d'apprentissage en k groupes (plis).
4. Pour chaque groupe unique :
   4.1. Attribution du rôle d'ensemble de validation à ce groupe.
   4.2. Attribution du rôle d'ensemble d'apprentissage réduit à l'ensemble formé par tous les autres groupes $(k-1)$.
   4.3. Ajustement du modèle sur l'ensemble d'apprentissage réduit.
   4.4. Validation du modèle et calcul du score sur l'ensemble de validation.
5. Évaluation du modèle à partir de la moyenne et de la variance des scores obtenus précédemment.
6. Le modèle final ainsi obtenu est ensuite entraîné sur l'ensemble d'apprentissage complet et évalué sur l'ensemble de test.

La valeur de k peut être n'importe quel nombre entier de telle sorte que chaque groupe de données d'apprentissage/validation soit suffisamment grand pour être statistiquement représentatif de l'ensemble des données. Les valeurs les plus courantes de k sont 5 et 10.

En ce qui concerne la taille des ensembles de validation et de test, on sait que l'ensemble de validation ne doit pas nécessairement être grand si on effectue une méthode robuste comme la CV. Quant à l'ensemble de test, il doit être de taille importante pour juger de la performance du modèle optimisé. Par conséquent, il est d'usage de diviser l'ensemble des données A en 70% pour l'apprentissage, 10% pour la validation et 20% pour le test. Ces chiffres sont souvent utilisés en première intention. Ils permettent de réaliser un premier essai sur un modèle donné mais doivent être ensuite ajustés en fonction de l'ensemble des données A et du cas d'étude.

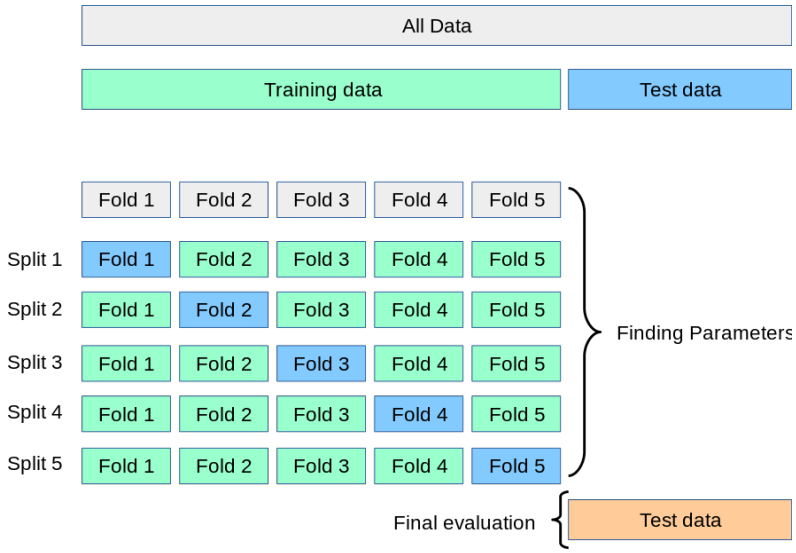


Figure 2.6. Validation croisée (Sckit-learn, 2011)

2.2.2 Catégories d'apprentissage automatique

Principaux types

Les systèmes d'apprentissage automatique peuvent être catégorisés selon plusieurs critères (Géron, 2022) (Figure 2.7) :

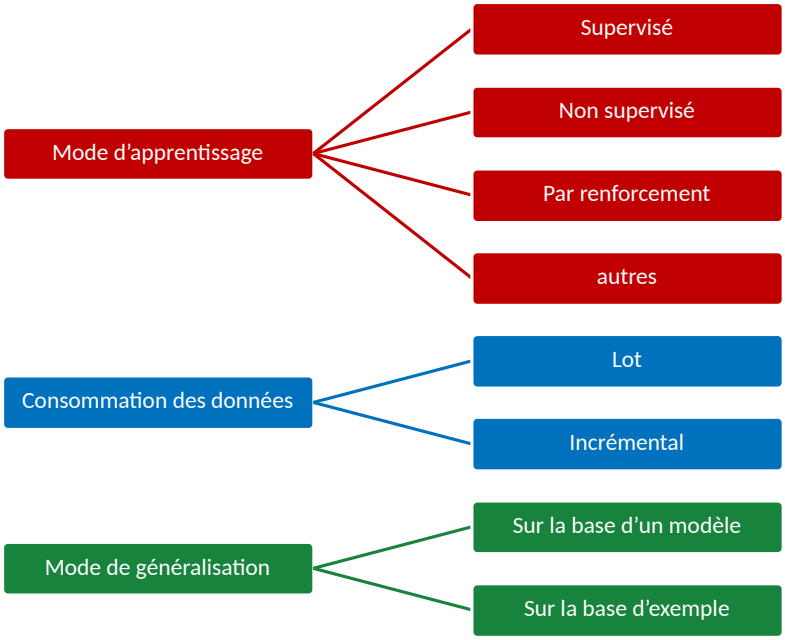


Figure 2.7. Les différents critères de catégorisation de l'apprentissage automatique

Par mode de généralisation : *selon que l'apprentissage se base sur l'exemple (instance-based learning) ou sur l'optimisation d'un modèle (model-based learning).*
Ce qui a été décrit précédemment dans le § 2.2.1 correspond au model-based learning : l'algorithme d'apprentissage détecte des tendances dans les données d'apprentissage et construit un modèle prédictif.
Une autre méthode d'apprentissage est l'instance-based learning : l'algorithme fonctionne en comparant simplement de nouveaux points de données à des points de données connus. Dit autrement, les nouvelles données sont prises en compte en fonction de leur similarité vis-à-vis des données passées, réputées exactes. Un exemple d'algorithme est le k-Nearest Neighbors (k-NN).

Par mode de consommation des données : *selon que l'apprentissage est réalisé par lot statique (ou groupé, Batch learning) ou sur flux continu (incrémental, Online Learning) Figure 2.8.* Historiquement, le Batch learning est le premier à apparaître pour traiter des données collectées préalablement : c'est un apprentissage avec des données statiques. Le modèle obtenu se contente d'appliquer sur les nouvelles données ce qu'il a déjà appris dans la phase d'entraînement. Puis, l'Online Learning a été conçu pour traiter les données arrivant de manière continue : l'algorithme apprend d'une manière incrémentale à partir d'un flux de données entrant (Salperwyck, 2013).

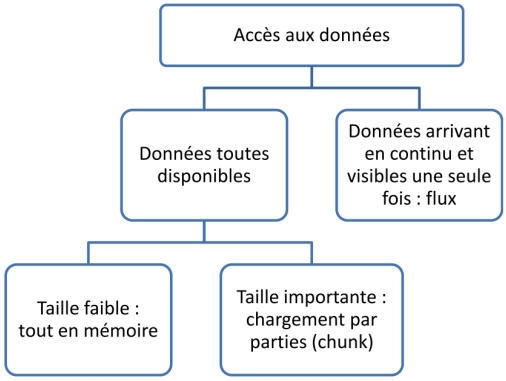


Figure 2.8. Les différents accès aux données pour l'apprentissage (Salperwyck, 2013)

Par mode d'apprentissage : *selon le type et la quantité de supervision humaine que reçoit l'algorithme pendant l'entraînement.*

Il existe de nombreuses catégories, mais seules les catégories principales sont abordées :

Supervisé (supervised) : l'algorithme d'apprentissage est entraîné en utilisant des données correctement étiquetées au préalable (classe, valeur continue...). En d'autres termes, la sortie cible (target output) y est connue et est injectée dans le modèle lors de la phase d'apprentissage.

Non supervisé (unsupervised) : les données d'apprentissage ne sont pas étiquetées (y est inconnu). L'algorithme essaie d'apprendre sans professeur. C'est un cas assez fréquent puisque l'obtention des données n'est pas évidente.

Par renforcement (reinforcement) : l'algorithme d'apprentissage peut observer l'environnement, sélectionner et effectuer des actions, et obtenir des récompenses en retour (ou des pénalités sous forme de récompenses négatives). Il doit ensuite apprendre par lui-même quelle est la meilleure stratégie pour obtenir la meilleure récompense au fil du temps. C'est typiquement le type d'apprentissage utilisé pour l'entraînement des robots pour marcher ou pour gagner à un jeu comme le jeu de Go (AlphaGo, Figure 2.1) : la machine apprend la stratégie gagnante en analysant des millions de parties, puis en jouant de nombreuses parties contre elle-même

La catégorisation par mode d'apprentissage est la plus utilisée. Dans ce qui suit, nous rentrons dans le détail des modes d'apprentissage supervisé et non supervisé.

Les applications de l'apprentissage automatique supervisé

L'apprentissage supervisé est utilisé dans deux types de problèmes : la Régression (Regression) et la Classification (Classification) (Figure 2.11).

Régression : technique servant à estimer ou à prédire la valeur d'un attribut numérique (la variable cible continue) en se fondant sur la valeur d'un ou de plusieurs autres caractéristiques.

Classification : technique qui permet de prédire si une donnée appartient à une classe discrète (catégorie) (Figure 2.9).

Il convient de noter que de nombreux modèles sont capables d'effectuer aussi bien de la régression que de la classification. On peut citer notamment : les Séparateurs à Vaste Marge SVM (Support Vector Machine), les arbres de décisions DT (Decision Tree), les réseaux de neurones artificiels ANN (Artificial Neural Network).

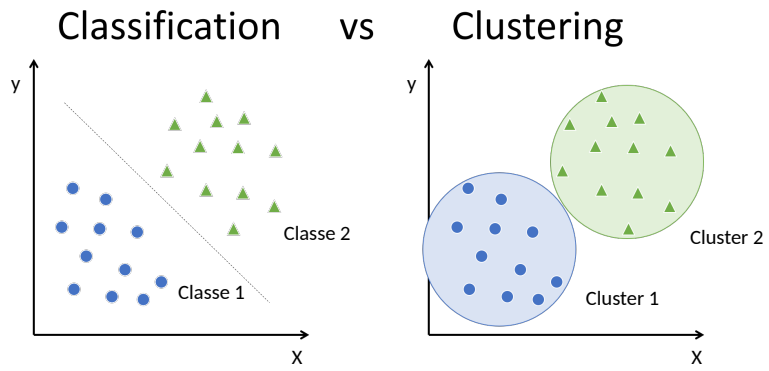


Figure 2.9. Différence entre la classification et le clustering, adapté de Keerthana (2022)

Les applications de l'apprentissage automatique non supervisé

L'apprentissage non supervisé est utilisé pour résoudre plusieurs types de problèmes : le regroupement de données (clustering), la réduction des dimensions, l'association et la détection d'anomalies.

Clustering : l'algorithme identifie les instances (observations de données) similaires et les affecte à des groupes (ou grappes, clusters).

Réduction des dimensions : l'algorithme réduit le nombre de paramètres en projetant les données issues d'un espace de grande dimension dans un espace de plus petite dimension. Cette technique permet de gérer ce que certains ont été jusqu'à nommer « le fléau des grandes dimensions » (the curse of dimensionality, (Bellmann, 1961)). Le but est de simplifier les données sans perdre trop d'informations. Cette opération est parfois cruciale en ML pour réduire la complexité du problème, accélérer l'apprentissage et améliorer les propriétés de stabilité et de robustesse des algorithmes. Parmi ces modèles on peut citer les auto-encodeurs et l'analyse en composantes principales PCA (Principal Component Analysis).

Association : l'algorithme utilise différentes règles pour trouver des relations entre les variables d'un ensemble de données. Ces méthodes sont fréquemment utilisées

pour répondre aux questions telles que "les clients qui ont acheté cet article ont également acheté . . . ".

Détection d'anomalies : le modèle obtenu permet d'identifier les données aberrantes (outliers) d'un jeu de données. Cela permet de nettoyer de façon automatisée le jeu de données et d'améliorer l'utilisation qui en est faite (amélioration de la fiabilité d'un modèle de prédiction par exemple). Au cours de l'apprentissage, l'algorithme est confronté à des instances normales, ce qui lui permet d'apprendre à les reconnaître ; ensuite, lorsqu'il voit une nouvelle instance, il peut dire si elle ressemble à une instance normale ou s'il s'agit d'une anomalie (Figure 2.10) (Chandola et al., 2009).

Une autre tâche très similaire est la détection de nouveauté (novelty detection) : elle vise à détecter les nouvelles instances qui semblent différentes de toutes les instances de l'ensemble d'apprentissage. Pour cela, il faut disposer d'un ensemble d'apprentissage très "propre", dépourvu de toute instance qu'on souhaite que l'algorithme détecte.

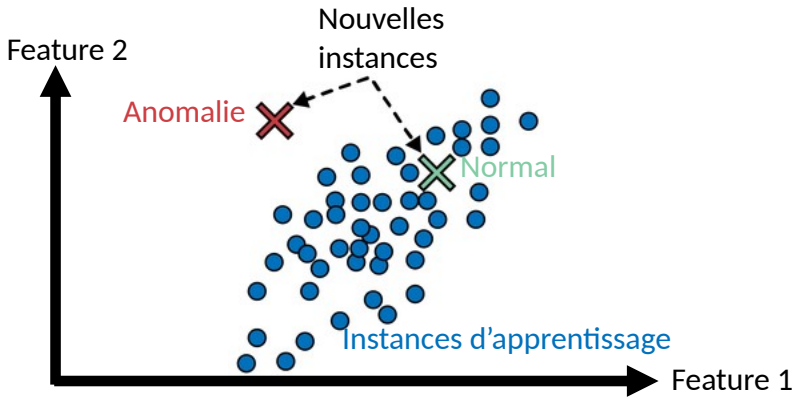


Figure 2.10. Détection d'anomalie (adaptée de Géron, 2022)

2.3 Algorithmes d'apprentissage automatique

Au cours des années, de nombreux algorithmes d'apprentissage automatique ont été développés. L'utilisation de ces algorithmes par des ingénieurs est devenue courante, mais il est crucial de les comprendre suffisamment pour optimiser les algorithmes de manière rigoureuse. Cette section a donc pour but de présenter de manière simplifiée certains algorithmes qui sont testés dans le cadre de cette thèse, sans toutefois fournir une description exhaustive. De plus, les mesures de performance des modèles les plus couramment utilisées sont présentées.

2.3.1 Introduction aux algorithmes d'apprentissage automatique

Les algorithmes d'apprentissage automatique peuvent être divisés en trois grandes catégories : les algorithmes classiques, les méthodes ensemblistes (ensemble methods) et

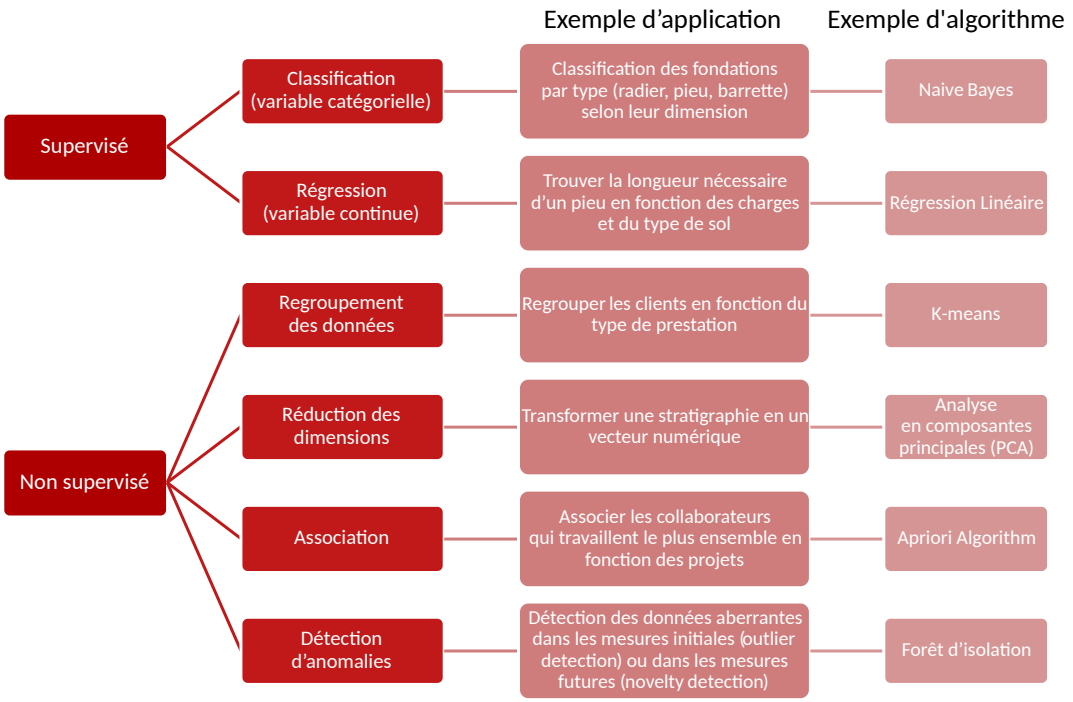


Figure 2.11. Différents exemples d'application et d'algorithmes des modèles supervisés et non supervisés de ML

les réseaux de neurones artificiels (artificial neural network ANN). Avant de décrire ces différentes catégories, un bref rappel historique du développement de ces méthodes est présenté. Il convient de noter que les hyperparamètres de chacun de ces algorithmes sont détaillés dans le § 6.2.

Historique

L'histoire des algorithmes d'apprentissage automatique remonte à plusieurs décennies. Le premier algorithme d'apprentissage automatique connu est le perceptron, qui a été développé en 1957 par Frank Rosenblatt (Figure 2.1). Au cours des années 1960 et 1970, de nombreux algorithmes d'apprentissage automatique ont été développés, notamment les algorithmes de réseaux de neurones, d'apprentissage profond et de classification bayésienne. Puis, entre 1980 et 1990, de nouveaux algorithmes ont été développés, tels que les réseaux convolutifs (LeCun et al., 1989), les arbres de décision DT (Quinlan, 1986), les forêts aléatoires RF (Breiman, 2001) et les Séparateurs à Vaste Marge SVM (Boser et al., 1992 ; Cortes et al., 1995). Au cours des années 2000 et 2010, l'apprentissage automatique a connu une croissance explosive en raison de l'augmentation des capacités de calcul et des avancées en matière de traitement des données. C'est l'ère de l'apprentissage profond DL (Deep Learning) qui a permis de résoudre des problèmes complexes en utilisant des réseaux de neurones profonds et sophistiqués. De nouveaux algorithmes continuent à être développés comme par exemple l'algorithme Extreme Gradient Boosting (XGBoost) (Chen et He, 2014) qui a rapidement gagné en popularité dans la communauté des data scientists

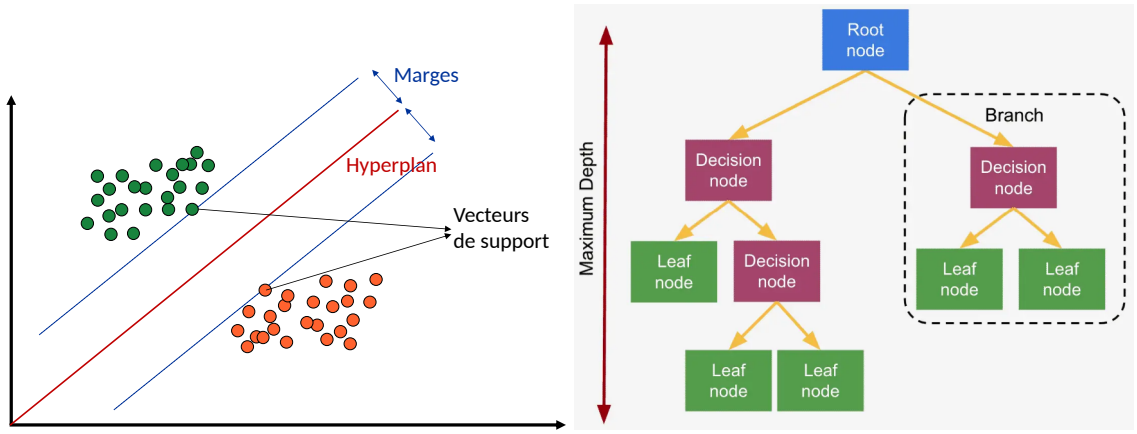
grâce à son succès dans de nombreux concours de machine learning, remportant souvent les premiers prix. Aujourd'hui, l'apprentissage automatique est largement utilisé dans l'industrie et le monde académique pour résoudre des problèmes complexes de traitement des données dans de nombreux domaines, notamment la finance, la santé, la recherche et le marketing.

Algorithmes classiques

Un algorithme classique est un algorithme d'apprentissage automatique considéré comme une référence ou un point de départ bien établi pour résoudre des problèmes similaires. En général, un algorithme classique est largement utilisé, bien compris, relativement simple à comprendre et à implémenter et a été rigoureusement validé à travers de nombreuses applications et tests.

Séparateur à vaste marge (SVM, Support Vector Machine) : SVM est un algorithme d'apprentissage automatique supervisé qui est utilisé pour résoudre des problèmes de classification et de régression (Boser et al., 1992 ; Cortes et al., 1995). L'idée de base est de trouver un hyperplan qui sépare de manière optimale les données en deux classes. Pour ce faire, l'algorithme cherche à maximiser la marge entre les exemples d'entraînement les plus proches de chaque classe, appelés vecteurs de support (support vector), et l'hyperplan de décision tout en minimisant l'erreur (Figure 2.12a). Dans le cas où les données ne sont pas linéairement séparables, SVM utilise une technique appelée « fonction de noyau » (kernel function) pour projeter les données dans un espace de dimension supérieure où elles pourront l'être.

Arbre de décision (DT, Decision Tree) : DT est un algorithme d'apprentissage automatique supervisé qui est utilisé pour résoudre des problèmes de classification et de régression (Quinlan, 1986). L'idée est de diviser récursivement l'espace de recherche en utilisant des règles de décision simples sur les différentes caractéristiques (features) des données. L'algorithme commence par le nœud racine (root node), qui représente l'ensemble des données d'entrée. À chaque nœud, l'algorithme sélectionne la caractéristique qui fournit la meilleure séparation possible des données vis-à-vis de la variable cible. Cette caractéristique est utilisée pour diviser les données en sous-ensembles plus petits, qui sont ensuite traités de manière récursive pour former des sous-arbres (branch). Le processus de division se poursuit jusqu'à ce que toutes les données d'un sous-ensemble soient classées dans une seule classe ou jusqu'à ce que les critères d'arrêt soient atteints (par exemple, un nombre maximum de profondeur de l'arbre (maximum depth)) (Figure 2.12b). L'arbre de décision résultant peut être interprété graphiquement et utilisé pour prédire la classe d'un nouvel exemple en suivant le chemin à travers l'arbre qui correspond aux caractéristiques de cet exemple.



(a) SVM, adapté de Premanand (2021) (b) DT (Chouinar, 2022)

Figure 2.12. Illustration du fonctionnement des algorithmes classiques SVM et DT

Réseaux de neurones

Les réseaux de neurones artificiels (ANN, Artificial Neural Network) sont des algorithmes d'apprentissage automatique supervisé, inspirés de la structure du cerveau humain. Ils sont utilisés pour résoudre des problèmes de classification, de régression, de reconnaissance de formes et de traitement du langage naturel, entre autres. Un réseau de neurones artificiels est composé d'un ensemble de nœuds interconnectés, appelés neurones, organisés en couches. Les neurones de la couche d'entrée reçoivent les données d'entrée, les neurones des couches cachées effectuent des calculs de traitement des données et les neurones de la couche de sortie renvoient les résultats. Chaque neurone effectue une fonction mathématique simple, généralement une somme pondérée des entrées suivie d'une fonction d'activation non linéaire. Les poids associés à chaque entrée sont ajustés pendant la phase d'apprentissage pour minimiser l'erreur de prédiction. Le processus d'apprentissage se fait par rétropropagation (backpropagation), qui consiste à calculer l'erreur de prédiction et à propager cette erreur à travers le réseau pour ajuster les poids des neurones de manière itérative jusqu'à ce que l'erreur soit minimisée.

Les réseaux de neurones peuvent être construits avec différentes architectures et fonctionnalités, telles que des réseaux de neurones convolutionnels (convolutional neural network) pour la reconnaissance d'images, des réseaux de neurones récurrents (recurrent neural network) pour la modélisation de séquences, ou des réseaux de neurones adverses (adversarial neural network) pour la génération de contenu créatif (Figure 2.13).

Méthodes ensemblistes

Les méthodes ensemblistes (ensemble methods) constituent une technique utilisée en apprentissage automatique qui combine plusieurs modèles de façon à ce qu'ils travaillent ensemble de manière plus efficace. Ces méthodes visent à améliorer la performance, la

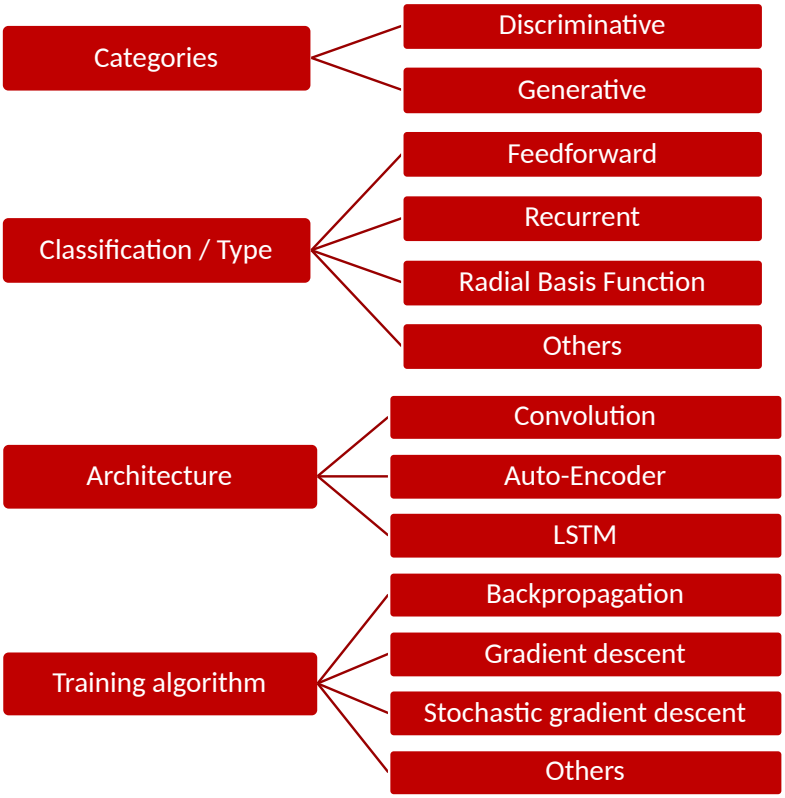


Figure 2.13. Les différents catégories, types, architectures et algorithmes d'entraînement des réseaux de neurones

robustesse et la stabilité des modèles de prédiction. La combinaison des modèles peut se faire avec différentes techniques parmi lesquelles on peut citer le bagging ou le boosting.

Bagging : Le bagging consiste à entraîner plusieurs algorithmes de manière indépendante sur des sous-ensembles de données aléatoires tirés de l'ensemble de données d'origine. Ces sous-ensembles sont générés de manière à ce qu'ils soient approximativement de la même taille et qu'ils contiennent des exemples choisis de manière aléatoire avec remise (c'est-à-dire qu'un exemple peut être sélectionné plusieurs fois dans le même sous-ensemble). Les prédictions de chaque algorithme sont ensuite combinées de manière à obtenir une prédiction finale.
Le bagging est souvent utilisé avec des algorithmes de forêts aléatoires (Random Forests RF), où chaque arbre de décision est entraîné sur un sous-ensemble de données aléatoire et les prédictions de chaque arbre sont combinées pour obtenir une prédiction finale. Cette technique peut améliorer la robustesse des modèles en réduisant le sur-apprentissage et en permettant de mieux gérer la variance des prédictions.

Boosting : Le concept derrière le boosting est de construire des modèles de façon successive, en commençant par un modèle de base et en ajoutant de nouveaux modèles qui « corrigent » les erreurs commises par les modèles précédents. Chaque modèle est construit en se concentrant sur les exemples qui ont été mal prédits par

les modèles précédents, ce qui permet d'améliorer progressivement la performance globale. Il existe plusieurs algorithmes de boosting, tels que l'algorithme AdaBoost (Adaptive Boosting) et l'algorithme XGBoost (eXtreme Gradient Boosting). Le boosting est souvent utilisé avec des algorithmes de forêts d'arbres de décision(Decision Trees DT), mais il peut également être utilisé avec d'autres types d'algorithmes.

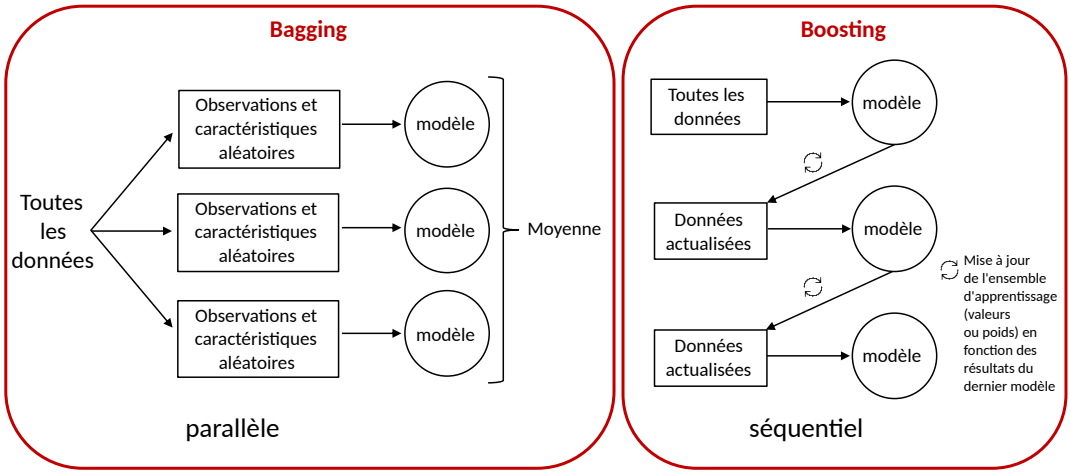


Figure 2.14. Apprentissage ensembliste : Bagging et Boosting

Dans ce qui suit, les algorithmes ensemblistes XGBoost, forêt aléatoire (Random Forest, RF) et forêt d'isolation (Isolation Forest, IF) sont introduits brièvement.

XGBoost (Extreme Gradient Boosting) : L'algorithme XGBoost est un algorithme d'apprentissage automatique supervisé, développé en 2014 par Chen et He (2014) pour améliorer les performances et la vitesse de l'algorithme Gradient Boosting Machine (GBM). Des améliorations significatives ont depuis été apportées à XGBoost, telles que la parallélisation et la distribution des calculs sur plusieurs nœuds, ce qui a permis une accélération significative des temps d'entraînement. De plus, des algorithmes de régularisation ont été introduits pour prévenir le sur-apprentissage, d'autre optimisant le choix des sous-échantillons pour une meilleure utilisation de la mémoire.

Forêt aléatoire (Random Forest, RF) : L'algorithme des forêts aléatoires est un algorithme d'apprentissage supervisé introduit en 2001 par Breiman (2001). C'est un modèle d'ensemble qui combine les prédictions de plusieurs arbres de décision pour produire des prédictions plus précises et plus robustes. Les arbres de décision individuels sont entraînés sur des sous-ensembles aléatoires des données d'entraînement et sur des sous-ensembles aléatoires des caractéristiques. Cette technique permet de réduire le surapprentissage et d'améliorer la généralisation du modèle. Les forêts aléatoires peuvent être utilisées pour la classification ou la régression, et sont particulièrement adaptées aux données de grande dimension et à haute cardinalité (grand nombre de caractéristiques avec de nombreuses valeurs uniques). Elles sont également résistantes aux valeurs aberrantes, ce qui en fait un choix populaire

pour les données bruitées. Les forêts aléatoires ont connu un grand succès et sont souvent considérées comme l'un des algorithmes d'apprentissage automatique les plus performants.

Forêt d'isolation (Isolation Forest, IF) : L'algorithme des forêts d'isolation est un algorithme de détection d'anomalies utilisé pour identifier les observations atypiques dans un ensemble de données. Il fonctionne en construisant plusieurs arbres de décision, appelés « arbres d'isolation », où chaque arbre est formé à partir d'une sous-sélection aléatoire des données d'entraînement. L'idée de base est de séparer les observations normales des observations atypiques en utilisant des arbres de décision. Plus précisément, les observations atypiques auront une profondeur d'arbre inférieure à celle des observations normales, car elles seront isolées plus rapidement dans les arbres. Enfin, une note d'isolement est attribuée à chaque observation en fonction de la profondeur de l'arbre où elle a été isolée. Les observations avec une note d'isolement plus faible sont considérées comme étant probablement des anomalies. Cet algorithme est bien adapté pour l'apprentissage non supervisé, en particulier donc pour les tâches de détection d'anomalies.

2.3.2 Mesures de performance

La qualité des prédictions d'un modèle détermine sa performance. Pour quantifier cette performance, l'erreur entre les valeurs prédites $\hat{y}$ et les valeurs réelles y est mesurée (§ 2.2.1). Par définition, plus l'erreur est faible, plus le modèle est considéré comme performant. Les fonctions de perte (loss functions) sont utilisées pour calculer l'erreur pour un seul point de données, tandis que les fonctions de coût (cost functions) sont utilisées pour calculer l'erreur sur l'ensemble du jeu de données. En régression, une fonction de perte est généralement formulée sous la forme $(y_i - \hat{y}_i)$, tandis qu'une fonction de coût est donnée par $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$ où n est la taille du jeu de données (Padhma, 2021). Le terme « mesure de performance » (performance measure) est également couramment utilisé. Il existe de nombreuses mesures de performance du modèle en fonction de la tâche considérée, qu'il s'agisse de régression, classification, clustering, etc. Dans cette partie, quelques mesures de performance pour les modèles de régression sont présentées avec leurs avantages et inconvénients.

Les principales mesures de performance pour la régression sont : R^2, MAE et MSE et la racine de ce dernier, RMSE. D'autres métriques moins utilisées sont QE, et MAPE, RAE, RRMSE, ME, Huber Loss, Log Cosh Loss et bien d'autres.

Coefficient de détermination R^2 (coefficient of determination) : Il existe plusieurs méthodes pour le calcul de R^2, qui permettent de mesurer la qualité de l'ajustement d'un modèle de régression aux données. Pour un modèle de régression linéaire simple, R^2 est égal au carré du coefficient de corrélation de Pearson R (Équation 2.1). En utilisant cette méthode, R^2 est obligatoirement un nombre positif compris entre 0 et 1. Cependant, d'autres formulations existent telles que l'Équation 2.2

et l'Équation 2.3 qui est également nommée qualité du calage goodness of fit. Ces formulations peuvent donner des valeurs négatives si la performance du modèle est médiocre. Il convient de noter que R^2 est très sensible aux valeurs aberrantes et peut augmenter de manière absurde lorsque des caractéristiques (features) sont ajoutées au modèle.

$$R = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})}} \tag{2.1}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{2.2}$$

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{2.3}$$

Erreur absolue moyenne (Mean Absolute Error MAE) : *également appelée erreur* L_1. Une faible MAE indique une bonne performance du modèle, car elle permet de connaître la distance absolue moyenne entre les prédictions et les mesures réelles. Contrairement à d'autres mesures de performance, la MAE est plutôt robuste, car elle est peu affectée par les valeurs aberrantes (outliers) (Trevisan, 2022). Cependant, étant donné que la MAE est une valeur absolue, elle ne permet pas de déterminer la direction de l'erreur (sous-estimation ($\hat{y} < y$) ou sur-estimation ($\hat{y} > y$)).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.4}$$

Erreur quadratique moyenne (Mean Squared Error MSE) : *également appelée erreur* L_2. Une faible valeur de MSE indique une bonne performance du modèle. Tout comme la MAE, la MSE détermine, en moyenne, à quel point la prédiction est proche de la valeur réelle et elle ne permet pas de déterminer la direction de l'erreur. Cependant, contrairement à la MAE, la MSE élève au carré les erreurs, ce qui donne plus de poids aux valeurs aberrantes. Ainsi, pour savoir si le modèle a effectué de mauvaises prédictions sur certains points, il est préférable d'utiliser la MSE plutôt que la MAE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.5}$$

Racine carrée de l'erreur quadratique moyenne (Root Mean Square Error RMSE) : La RMSE est calculée en prenant la racine carrée de la MSE, ce qui permet d'exprimer l'erreur dans la même unité que la variable cible y. Une valeur faible de RMSE indique une bonne performance du modèle. La RMSE a tendance à amplifier davantage les grandes erreurs que les petites, ce qui peut conduire à une évaluation biaisée de la performance du modèle.

$$RMSE = \sqrt{MSE} \tag{2.6}$$

Erreur quantile (Quantile Error or Quantile Loss QE) : Une valeur faible de QE indique une bonne performance du modèle. Contrairement à la MAE, la QE est en mesure de pénaliser spécifiquement la sous-estimation ou la sur-estimation de l'erreur. Cela est bénéfique dans certaines applications où l'une ou l'autre direction peut avoir des conséquences plus critiques.

$$QE = \frac{1}{n}\Big[\sum_{i|_{(y_i<\hat{y}_i)}}(1-\gamma)|y_i-\hat{y}_i| + \sum_{i|_{(y_i\geq\hat{y}_i)}}\gamma|y_i-\hat{y}_i|\Big] \tag{2.7}$$

La valeur de γ est à choisir en fonction de la pénalité à donner. Pour $\gamma = 0.5$, on retrouve la formule initiale de la MAE.

Erreur relative absolue moyenne (Mean Absolute Percentage Error MAPE) : également appelée erreur relative moyenne (Mean Relative Error MRE). La MAPE prend en compte la distribution des erreurs de prédiction en pourcentage. Plus précisément, il s'agit de la moyenne des pourcentages des différences absolues entre les valeurs prédites et les valeurs réelles. Son usage est comparable à la MAE, mais elle exprimée en [%].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}}{y_i}\right| \tag{2.8}$$

Conclusion

Ce chapitre comportait trois parties visant à démystifier l'Intelligence Artificielle (IA) à travers une description de son historique, des équations de base de l'apprentissage automatique et de la présentation des algorithmes d'apprentissage les plus populaires pour des applications similaires à la prédiction des tassements induits par le creusement des tunnels.

Dans un premier temps, un bref historique de l'IA a été présenté, ainsi que les définitions de base de l'Intelligence Artificielle, de l'apprentissage automatique et du Big Data. Ensuite, l'importance cruciale des données pour les applications d'apprentissage automatique a été mise en évidence, en soulignant les travaux de collecte, traitement et stockage des données. La question de l'empreinte carbone de la donnée a également été brièvement abordée.

Dans un second temps, les concepts de base de l'apprentissage automatique ont été expliqués selon une vision d'ingénieur plutôt que de mathématicien. Ainsi, les notions d'algorithme d'apprentissage, de modèle obtenu avec ses paramètres, de fonction de coût pour évaluer la performance des algorithmes, de division des données pour l'entraînement et l'évaluation, de validation croisée ainsi que les risques de surajustement (overfitting) et de sous-ajustement ont été présentées. Les différentes catégories d'apprentissage automatique selon différents critères ont également été introduites (Figure 2.7).

Enfin, les algorithmes d'apprentissage automatique qui seront utilisés dans le cadre de cette thèse ont été présentés selon les trois catégories suivantes : modèles classiques

(SVM et DT), modèles ensemblistes (XGBoost, RF, IF) et réseaux de neurones (ANN). Les équations des métriques de performance pour mesurer la qualité des prédictions d'un modèle de régression ont également été introduites.

En résumé, ce chapitre représente une étape importante dans la compréhension du monde de l'Intelligence Artificielle et de l'apprentissage automatique adaptée à l'ingénieur géotechnicien, permettant ainsi d'explorer ces outils pour la prédiction des tassements induits par le creusement des tunnels.

Le chapitre suivant sera consacré à l'état de l'art de l'application des outils d'apprentissage automatique pour la prédiction des tassements. Il constitue également une introduction à la méthodologie de travail adoptée dans ces travaux de thèse.

ÉTAT DE L'ART DE LA PRÉVISION DES TASSEMENTS À L'AIDE D'OUTILS D'APPRENTISSAGE AUTOMATIQUE

3

Introduction

La prévision des tassements induits par le creusement des tunnels est essentielle pour évaluer la vulnérabilité des structures avoisinantes au creusement, et prévenir les risques de dommages. Toutefois, les méthodes actuelles, principalement basées sur des approches empiriques et numériques, présentent des limites liées aux hypothèses simplificatrices sous-jacentes et/ou à leur temps de calcul (§ 1.3). L'accessibilité des méthodes d'Intelligence Artificielle (IA), ainsi que la disponibilité de données abondantes collectées sur les sites de creusement de tunnels, ouvrent la voie à de nouvelles approches de prévision des tassements. L'utilisation des outils d'apprentissage automatique pour la prévision des tassements n'est cependant pas une thématique nouvelle de la recherche dans ce domaine. Les premières tentatives datent de 1998 par Shi et al. (1998) qui ont utilisé des réseaux de neurones pour la prévision des tassements. Cependant, les études précédentes ont été entravées par plusieurs obstacles, tels que la rareté des données, la nécessité d'une connaissance approfondie des algorithmes d'apprentissage, ainsi que l'attente de l'émergence d'algorithmes plus performants pour ce type d'application, tels que les forêts aléatoires (RF) en 2001 et les XGBoost en 2014.

L'objectif de ce chapitre est de dresser l'état de l'art des approches d'apprentissage automatique pour la prévision des tassements en se basant sur 31 publications sur le sujet (Table 3.1). Une attention particulière est portée à la taille des ensembles de données, au choix des algorithmes, des mesures de performance, des caractéristiques et des variables cibles. De plus, une considération importante est accordée à la représentation du sol dans les caractéristiques des algorithmes d'apprentissage. Ce chapitre propose également une introduction à la méthodologie de travail adoptée dans le cadre de cette thèse à travers la description des différentes étapes d'un exercice d'apprentissage automatique.

3.1 Recours à l'apprentissage automatique en géotechnique

Un état des lieux de l'application des outils d'apprentissage automatique dans le domaine de la géotechnique est établi. Ensuite, une vue d'ensemble de la prévision des tassements induits par l'excavation au tunnelier à l'aide de l'apprentissage automatique est présenté.

3.1.1 Panorama général

Depuis la fin des années 80, l'Intelligence Artificielle a suscité un intérêt croissant dans le domaine de la géotechnique. Parmi les premières applications, on peut citer l'utilisation de la logique floue (fuzzy logic) pour l'interprétation des données issues d'un pénétromètre CPT (Mullarkey et Fenves, 1986). Depuis, l'utilisation de l'IA dans les nombreux sous-domaines de la géotechnique a connu une augmentation exponentielle (Figure 3.1).

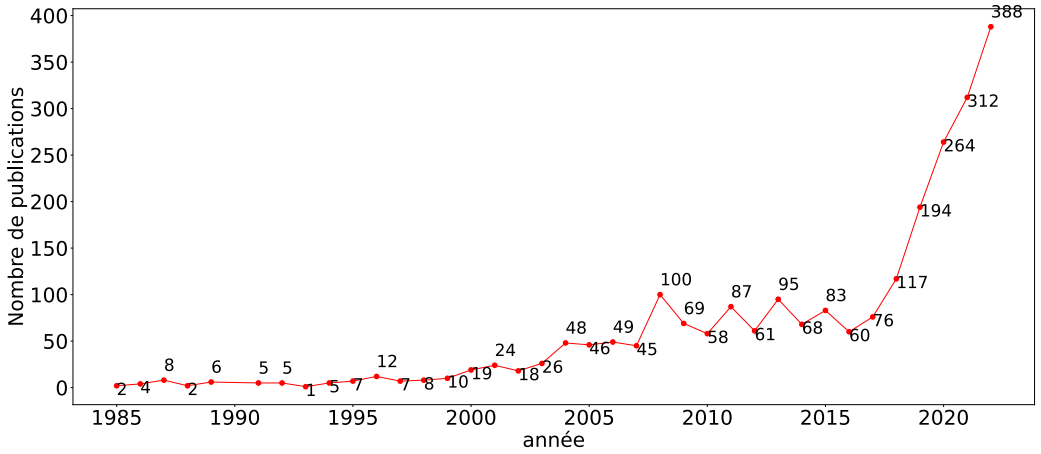


Figure 3.1. Nombre de documents publiés dans le domaine de la géotechnique et de l'IA jusqu'en 2022. Liste obtenue à l'aide de la requête sur la base de données de Scopus : (TITLE-ABS-KEY (("soil mechanic*" OR geotechni* OR geomechani*) AND ("artificial intelligence" OR "AI" OR "Machine Learning" OR "Soft Computing*" OR "deep learning" OR "neural network*")))

Parmi les domaines ou applications qui ont exploré l'utilisation de l'IA, on peut citer les sols gelés et propriétés thermiques des sols, la mécanique des roches, la caractérisation des sols et des roches, la caractérisation des sols de fondation et chaussées, les glissements de terrain et la stabilité des pentes, la liquéfaction des sols, les fondations superficielles et sur pieux, les tunnels et tunneliers, les barrages et les sols non saturés. L'état de l'art de l'application de l'IA dans ces nombreux domaines est établi par Baghbani et al. (2022) à travers l'étude de 1235 publications (Figure 3.2), ainsi que par Jong et al. (2021) et Ebid (2021). En outre, Zhang et al. (2021b) ont décrit spécifiquement l'état de l'art de l'application de l'apprentissage profond (Deep Learning) dans l'ingénierie géotechnique, tandis que Zhang et al. (2020b) se sont intéressés à l'état de l'art de l'application de l'IA aux tunnels.

L'état de l'art révèle que les réseaux de neurones artificiels sont le modèle le plus utilisé en géotechnique, représentant une part d'environ 50% des publications concernées (Baghbani et al., 2022 ; Ebid, 2021). Cela se comprend par le fait que c'est une des méthodes les plus anciennes. Les trois domaines qui se sont le plus intéressés par l'IA sont la mécanique des roches, les glissements de terrain et liquéfaction des sols et les tunnels et tunneliers (Figure 3.2). Selon les résultats de l'étude menée par Baghbani et al. (2022), il a été mis en évidence que l'efficacité et la précision des résultats obtenus à partir de ces

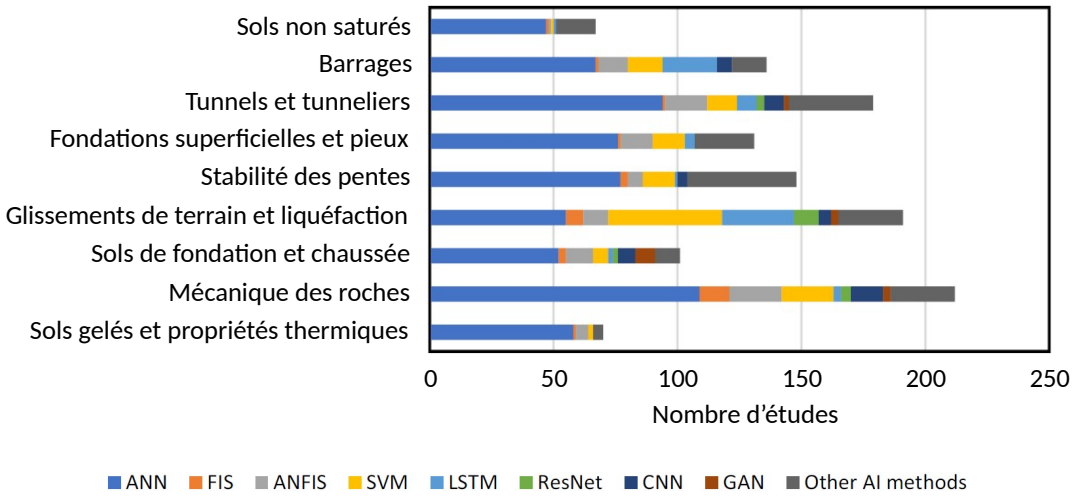


Figure 3.2. Répartition de l'utilisation de différentes techniques d'IA dans l'ingénierie géotechnique dans neuf domaines d'application en se basant sur 1235 articles publiés (adaptée de Baghbani et al., 2022)

méthodes dépendent fortement de facteurs tels que la taille de l'ensemble de données, le type de données traitées et la sélection de paramètres d'entrée pertinents. De plus, Ebid (2021) a souligné que les techniques présentées dans la recherche sont complexes à mettre en œuvre et nécessitent une adaptation par des experts métiers avant de pouvoir être appliquées dans des contextes réels.

3.1.2 Cas de la prévision des tassements induits par le creusement de tunnels

La prévision des tassements induits par le creusement des tunnels est cruciale pour prévenir les dommages aux structures avoisinantes. Les méthodes couramment utilisées, telles que les approches empiriques ou numériques, présentent certaines limitations (§ 1.3). La surveillance des déplacements du sol génère de grandes quantités de données spatiales et temporelles. Cette grande quantité de données, qui constitue une relative nouveauté dans la pratique géotechnique, laisse entrevoir de nouvelles perspectives quant à l'utilisation de méthodes de prévision à l'aide d'algorithmes d'apprentissage automatique. L'utilisation de ces algorithmes semblent être une méthode particulièrement adaptée et potentiellement précise pour estimer le tassement induit par le creusement des tunnels grâce à leur capacité à identifier les meilleures relations entre les différents paramètres (Huat et al., 2023).

Depuis la première tentative de prévision des tassements induits par le creusement des tunnels à l'aide d'algorithmes d'apprentissage automatique (Shi et al., 1998), plusieurs études ont été publiées au fil des années comme le montre la Figure 3.3. Cette figure met en évidence que ce sujet est particulièrement d'actualité, avec un pic dans le nombre de publications en 2022. De plus, on note que c'est la Chine et l'Iran qui ont le plus grand nombre de publications à ce sujet, avec environ 8 articles chacun à date. Ce nombre est

étroitement lié à la disponibilité de données générées par des projets de tunnels dans chaque pays. A titre d'exemple, le creusement des lignes de métro à Changsha en Chine et à Karaj en Iran a fait l'objet des études de Chen et al., 2019a ; Chen et al., 2019b ; Zhang et al., 2020a et de Hajihassani et al., 2020 ; Hasanipanah et al., 2016 ; Marto et al., 2012, respectivement.

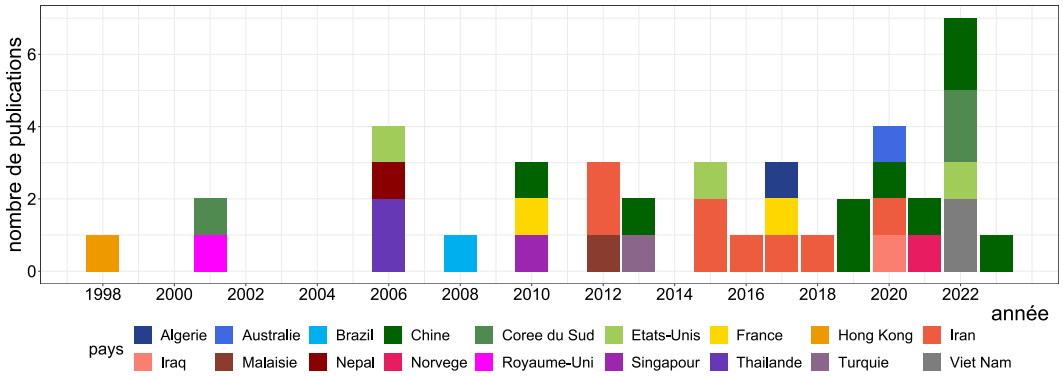


Figure 3.3. Évolution du nombre de publications traitant de la prévision des tassements causés par le creusement des tunnels à l'aide d'algorithmes d'apprentissage automatique en fonction du temps et répartition par pays d'étude (double compte en cas de publications en collaboration internationale)

La disponibilité de grandes quantités de données est un avantage crucial car les algorithmes d'apprentissage automatique nécessitent une quantité importante de données pour s'entraîner correctement et fournir des résultats fiables. Par conséquent, il est légitime de s'interroger sur la fiabilité des résultats des articles qui appliquent ce type d'algorithmes à de petits ensembles de données Table 3.1. Liu et al. (2022) ont mené une étude sur l'efficacité de la prévision des tassements à l'aide d'algorithmes d'apprentissage automatique avec des ensembles de données de petite taille. Il convient de noter que leur ensemble de données contient tout de même 187 valeurs. Selon leurs résultats, les algorithmes d'apprentissage automatique peuvent être utilisés avec succès pour prédire les tassements, même si la taille de l'ensemble de données disponible est petite. Toutefois, cette conclusion doit être vérifiée par des études supplémentaires portant sur des applications concrètes, ce qui fera l'objet d'une section de cette thèse (§ 6.2.2).

3.2 Analyse exploratoire des données

L'analyse exploratoire des données (exploratory data analysis, EDA) est la première étape de l'analyse de données et vise à connaître et comprendre les données avant de passer à des analyses plus sophistiquées. L'EDA comprend plusieurs parties, notamment la visualisation, l'exploration, le nettoyage et l'analyse statistique des données. Elle consiste ainsi à évaluer des quantités statistiques de base, à identifier des régularités, des motifs et des relations potentielles entre variables, ainsi qu'à détecter d'éventuelles anomalies, telles que des données aberrantes, en utilisant des méthodes visuelles.

Table 3.1. État de l'art sur la taille des jeux de données et les algorithmes d'apprentissage automatique utilisés (tout type de creusement)

Référence	Jeu de données	Algorithmes
Wang et al. (2013)	661	wsRVM
Su et al. (2022)	533	BPNN, SVM, XGBoost
Boubou et al. (2010)	432	BPNN
Bouayad et Emeriault (2017)	432	PCA-ANFIS
Shi et al. (1998)	356	BPNN
Zhou et al. (2023)	323	XGBoost, EN, RF
Mahmoodzadeh et al. (2020)	300	k-NN, SVM, DT, GP, DL, LSTM
Zhang et al. (2020a)	294	BPNN, GRNN, ELM, SVM, RF, (PSO)
Kim et al. (2022b)	253	RF
Kim et al. (2022a)	253	SVM, RF, GBM, LGBM, XGBoost
Ocak et Seker (2013)	230	BPNN, SVM, GP
Chen et al. (2019a)	200	BPNN, RBFNN, GRNN,
Chen et al. (2019b)	200	BPNN, GRNN, ELM, SVM, RF
Liu et al. (2022)	187	BPNN, MLP, SVM, DT, RF, GBM
Marto et al. (2012)	160	BPNN
Goh et Hefney (2010)	148	BPNN
Zhang et al. (2021a)	148	BPNN, MARS, SVM, XGBoost
Hasanipanah et al. (2016)	143	PSO-ANN
Hajihassani et al. (2020)	123	PSO-ANN
Kim et al. (2001)	113	BPNN
Chen et al. (2022)	101	BPNN, RF, XGBoost
Santos et Celestino (2008)	81	BPNN
Ahangari et al. (2015)	53	ANFIS, GEP
Suwansawat et Einstein (2006)	49	BPNN
Pourtaghi et Lotfollahi-Yaghin (2012)	49	WNN
Kohestani et al. (2017)	49	BPNN, WNN, RF
Moeinossadat et al. (2018)	41	NGS, ANFIS, GEP
Qiao et al. (2010)	41	BPNN
Neaupane et Adhikari (2006)	40	BPNN
Dindarloo et Siami-Irdemoosa (2015)	34	DT
Mohammadi et al. (2015)	17	BPNN, MLP

3.2.1 Exploration, nettoyage et stockage des données

L'exploration consiste à examiner les données pour en comprendre les caractéristiques principales. C'est une analyse préliminaire où l'on explore, on l'on peut chercher à confirmer des intuitions, ou enfin à faire émerger des concepts. Cela peut inclure l'identification des tendances, des modèles et des relations dans les données. La visualisation est une technique qui utilise des graphiques et des diagrammes pour représenter les données de manière claire et facilement compréhensible. Elle peut être utile pour mettre en évidence des tendances cachées, des relations ou des anomalies dans les données qui pourraient être difficiles à détecter autrement. À cette étape, un nettoyage des données est souvent nécessaire pour éliminer les incohérences ou les anomalies telles que les données aberrantes (outliers) ou les valeurs manquantes. En effet, la qualité des données a un impact direct sur la performance des algorithmes d'apprentissage automatique. Il est donc important de s'assurer que les données utilisées ultérieurement sont structurées, unifiées,

propres et interopérables (§ 2.1.2).

En ce qui concerne le stockage des données, il est recommandé de stocker les données nettoyées ainsi que les données brutes originales pour conserver toutes les informations et pour pouvoir effectuer des analyses supplémentaires si nécessaire. Le choix entre les tableurs et une base de données dépend de la taille des données et de la complexité des relations et ramifications entre elles (§ 2.1.2). Pour les projets de tunnel, la création de bases de données a déjà été abordée pour des projets tels que la Jubilee Line (Londres) (Withers et al., 2000) et l'Egnatia Highway (Grèce) (Marinos et al., 2013), ainsi que pour des tunnels de 100 km excavés par forage, dynamitage ou tunnelier à Montréal (Leroux et Campeau, 2018). Une base de données doit avoir une structure logique et robuste pour permettre des requêtes rapides et intuitives. Marinos et al. (2013) proposent une base de données formée par trois catégories principales : les mesures d'auscultation en surface, les paramètres de creusement du tunnelier et les données géologiques et géotechniques ; A ces trois catégories s'ajoute une catégorie secondaire contenant les tables d'assistance : abréviations, symboles et unités. Cette structure explicite facilite l'utilisation des données dans les étapes suivantes.

3.2.2 Analyses statistiques

L'analyse statistique est une méthode courante pour extraire des informations utiles à partir de données. Elle consiste à utiliser des outils statistiques pour étudier les relations entre différentes variables et évaluer la distribution des données. Il existe deux types d'analyses statistiques couramment utilisés : l'analyse univariée et l'analyse multivariée.

L'analyse univariée est une technique statistique qui examine une seule variable à la fois. Elle permet d'évaluer la distribution de cette variable. Elle peut également être utile pour détecter des valeurs aberrantes ou des données manquantes. La technique d'analyse univariée la plus courante est de visualiser la distribution des données sur un histogrammes (Kim et al., 2022a ; Kim et al., 2022b).

L'analyse multivariée est une méthode plus puissante pour l'analyse des données, puisqu'elle consiste en l'étude simultanée des relations entre deux ou plusieurs variables, en déterminant les corrélations et les dépendances entre ces variables. Les exemples de techniques d'analyse multivariée incluent le calcul du coefficient de corrélation de Pearson entre les variables (Équation 2.1), qui mesure la force de la relation linéaire entre deux variables, et l'utilisation de cartes de chaleur (heatmaps) pour visualiser les corrélations entre un grand nombre de variables prises deux à deux (Chen et al., 2019b ; Liu et al., 2022) (Figure 3.5).

Un exemple concret est présenté dans la Figure 3.4, qui donne les coefficients de corrélation entre les paramètres d'entrée et le tassement maximal. On constate que la corrélation n'est pas généralisable, mais dépend plutôt de l'ensemble des données. Néanmoins, quelques corrélations sont clairement visibles, telles que celles entre le tassement maximal et, d'une part, la poussée totale (P_{totale} [kN]) et, d'autre part, le couple de rotation de la roue de coupe (M_{RDC} [kN.m]).

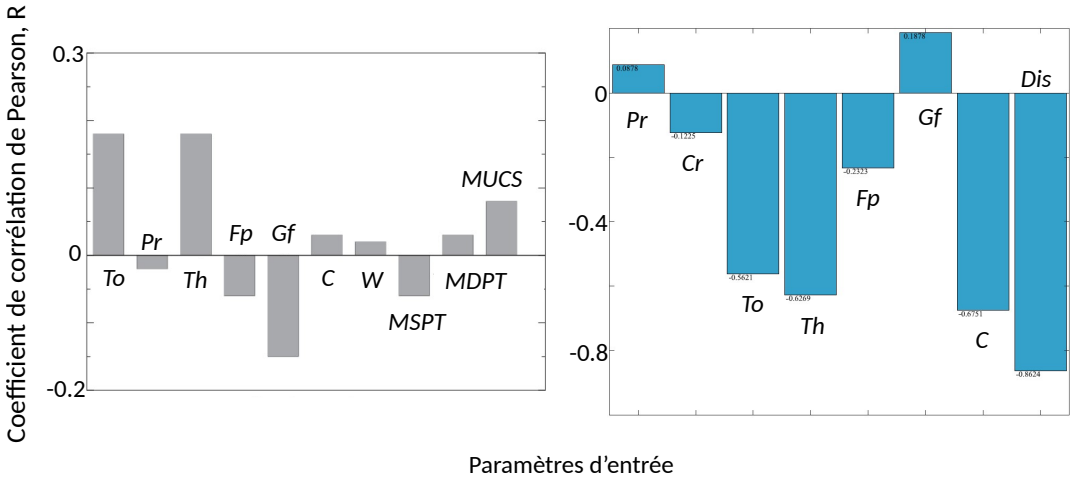


Figure 3.4. Coefficient de corrélation de Pearson entre des paramètres d'entrée et le tassement maximal observé. Figures issues des travaux de Chen et al. (2019a) à gauche et Zhou et al. (2023) à droite.
Légende : To = M_{RDC} [kN.m], Pr = $V_{tunnelier}$ [mm/min], Th = P_{totale} [kN], Fp = P_{front} [bar], Gf = $V_{mortier}$ [m^3], C [m], W = p_{nappe} [m], MSPT = MSPT, MDPT = MDPT, MUCS = MUCS, Cr = V_{RDC} [tour/min], Dis = d_{front}

3.3 Ingénierie des caractéristiques

L'ingénierie des caractéristiques (feature engineering) est une étape essentielle dans le processus de préparation des données pour l'apprentissage automatique, cela afin de garantir la qualité du modèle et la précision des prévisions. C'est un processus qui consiste à sélectionner les paramètres les plus pertinents (feature selection), à extraire les caractéristiques les plus importantes (feature extraction) et, éventuellement, à mettre à l'échelle ces caractéristiques pour les préparer à l'apprentissage automatique (feature scaling).

3.3.1 Sélection des caractéristiques

Après avoir effectué des analyses statistiques, il est possible de sélectionner les paramètres d'entrée les plus pertinents pour le modèle (feature selection). Cette sélection se fait en se basant sur : l'expertise métier, la corrélation des paramètres avec la variable de sortie, les formules empiriques (Neaupane et Adhikari, 2006), le retour d'expérience des modèles numériques (Wang et al., 2013) ou encore l'état de l'art.

Pour la prévision des tassements induits par le creusement des tunnels, les paramètres d'entrée sont généralement classés en trois catégories : la géométrie du tunnel, les conditions géologiques et les paramètres de creusement (Suwansawat et Einstein, 2006). Néanmoins, quelques études se sont passées des conditions géologiques (Boubou et al., 2010 ; Zhou et al., 2023). D'autres études portant sur des méthodes de creusement traditionnel (comme par exemple la méthode NATM) ne prennent pas en compte de paramètres de creusement (Ahangari et al., 2015 ; Hasanipanah et al., 2016 ; Marto et al., 2012 ; Mohammadi et al., 2015).

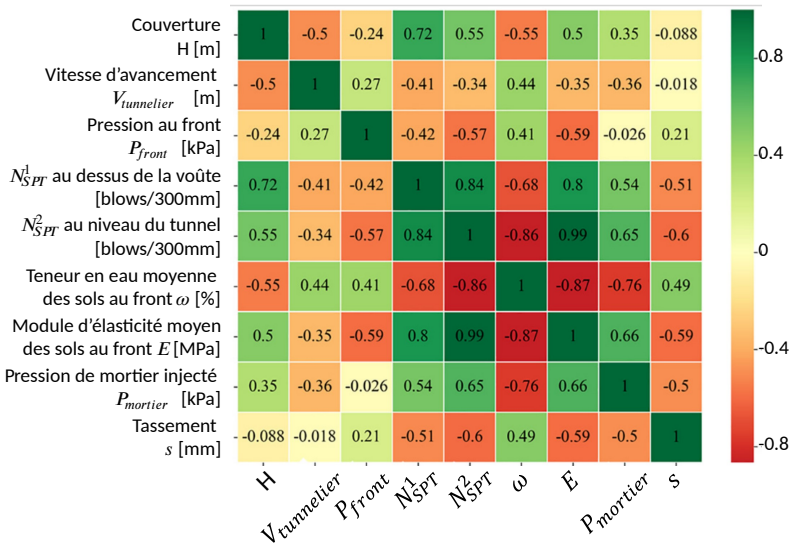


Figure 3.5. Carte de chaleur (heatmap) obtenue par Zhang et al. (2021a) pour leur ensemble de données

En se limitant aux études portant sur le creusement de tunnels au tunnelier (quel qu'en soit le type), les paramètres d'entrée les plus couramment considérés sont la couverture du tunnel, la vitesse d'avancement, la pression au front, ainsi que le volume et la pression d'injection du mortier à l'arrière de la jupe (Figure 3.6). Il convient de mentionner que la distance au front est toujours prise comme caractéristique pour la prévision de s_{long} et la distance à l'axe pour la prévision de s^*.

Quelques études ont testé des méthodes comme la sélection récursive arrière (backward selection) (Boubou et al., 2010) ou la sélection récursive avant (forward selection) (Ocak et Seker, 2013) ce qui leur permet de réduire le nombre de caractéristiques à introduire dans le modèle final. La backward selection en apprentissage automatique consiste à entraîner un modèle avec toutes les caractéristiques disponibles, puis à éliminer progressivement les caractéristiques les moins importantes jusqu'à ce qu'un modèle suffisamment performant soit obtenu. La forward selection, à l'inverse, commence avec un modèle vide et ajoute progressivement les caractéristiques les plus importantes.

Une autre méthode, testée par Kim et al. (2022b), se base sur une sélection des caractéristiques pilotée par les données (data-driven feature selection) et consiste à calculer trois indices : (Predictive power score, Mutual information et Feature importance) pour évaluer l'importance de chacune des caractéristiques vis-à-vis du tassement prévu.

La technique d'analyse de l'importance des caractéristiques (feature importance FI) est utilisée dans plusieurs études pour évaluer le choix des caractéristiques (Su et al., 2022 ; Zhang et al., 2020a ; Zhang et al., 2021a). Liu et al. (2022) ont entraîné un premier modèle de réseaux de neurones artificiels, puis ont utilisé une technique d'analyse d'importance par permutation des caractéristiques (feature permutation importance FPI). Cette méthode consiste à mesurer la réduction de score d'un modèle lorsque chaque caractéristique est mélangée de manière aléatoire. Cela permet ainsi de déterminer la contribution de chaque caractéristique à la performance globale du modèle (Breiman, 2001). Les caractéristiques

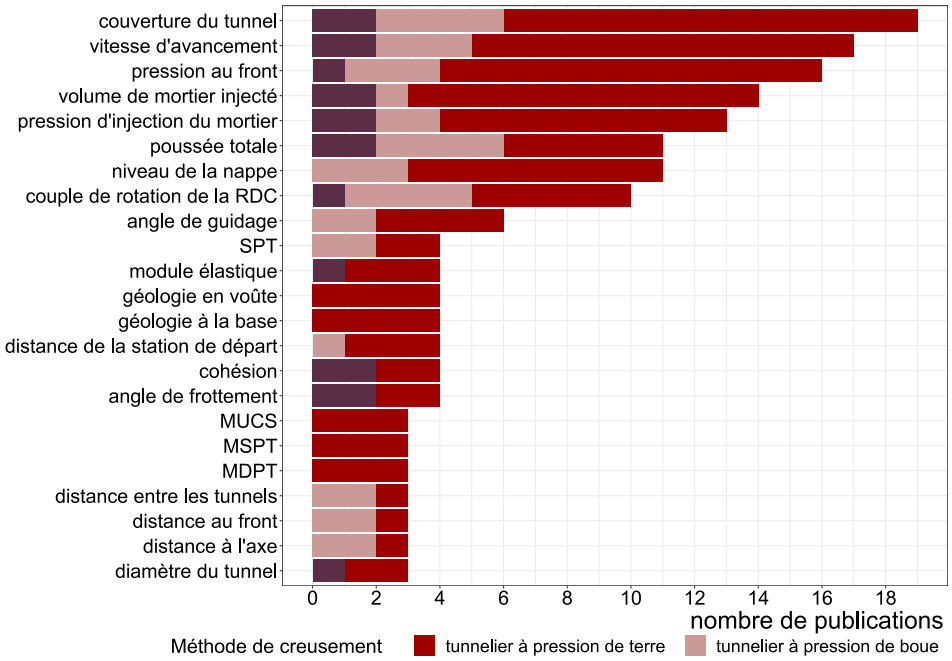


Figure 3.6. Les principaux paramètres d'entrée utilisés dans l'état de l'art pour la prévision des tassements causés par le creusement au tunnelier uniquement

les plus significatives ont ensuite été sélectionnées et définies comme caractéristiques du réseau de neurones final, ce qui a permis de réduire la complexité de l'architecture.

3.3.2 Extraction des caractéristiques

La sélection des paramètres d'entrée est suivie par la tâche cruciale d'extraction des caractéristiques. Cette dernière consiste à combiner les caractéristiques existantes pour produire une caractéristique plus utile et plus informative qui permettra d'obtenir une meilleure prévision du résultat. La qualité du traitement de cette tâche repose sur une bonne connaissance du problème métier. En tant que processus de décision subjectif, cela nécessite de l'intuition et de la créativité. La transformation des caractéristiques peut se faire de différentes manières, dont voici quelques exemples :

– Combiner plusieurs paramètres d'entrée en appliquant des transformations mathématiques telles que l'addition, la soustraction, la multiplication, la division, la suppression ou encore la combinaison Par exemple, pour accorder un poids plus important aux données proches de l'axe du tunnel, Boubou et al. (2010) ont pris comme paramètre d'entrée $exp(\frac{-d_{axe}^2}{H^2})$ (avec d_{axe} et H [m]), ce qui a très légèrement amélioré les résultats.

– Utiliser des transformations mathématiques. Certains algorithmes d'apprentissage ont du mal à détecter des structurations lorsque les distributions des variables sont fortement dissymétriques. Pour détecter une variable dissymétrique, il suffit de tracer son histogramme pour visualiser sa distribution : on verra sur des cas concrets que les valeurs s'étendent souvent bien plus loin d'un côté de la médiane

que de l'autre. Le but est alors de transformer les variables dissymétriques de façon à avoir des distributions plus proche d'une courbe gaussienne (normalisation). Cette transformation peut être obtenue par exemple en prenant le logarithme ou la racine carrée de la variable, en élevant la variable à une puissance (power transform) ou encore en utilisant la transformée de Box-Cox, généralisable à toute distribution continue. La transformation logarithmique a été testée par Ahangari et al. (2015) ce qui a permis d'enlever un niveau de transformation « évident » à la source et donc réduire la complexité de sa résolution.

– Réduire la dimension du problème. Pour réussir à entraîner un modèle avec un grand nombre de caractéristiques, il faut avoir suffisamment de données. Dans des applications comme la prévision des tassements, la quantité de données est en général très limitée alors même que le nombre de paramètres d'entrée très grand, notamment pour ce qui est de décrire la stratigraphie (cf. paragraphe suivant). Pour ce faire, des méthodes comme l'analyse en composantes principales (Principal Component Analysis, PCA) peuvent être appliquées. Bouayad et Emeriault (2017) ont testé cette approche pour passer de 15 paramètres d'entrée (paramètres de tunnelier et épaisseur des couches de sol) à 7 caractéristiques.

– Transformer des variables catégorielles en variables numériques. Ce type de tâche est souvent essentiel car de nombreux modèles d'apprentissage automatique ne peuvent pas traiter directement les variables catégorielles. Plusieurs techniques existent pour réaliser cette transformation, notamment l'encodage « un parmi n » (one-hot encoding), qui consiste à représenter chaque catégorie de la variable sous forme d'un vecteur binaire. Dans ce vecteur, toutes les entrées sont nulles, sauf une qui correspond à la catégorie de la variable. On peut citer l'exemple de Suwansawat et Einstein (2006) qui ont utilisé le one-hot encoder pour définir la géologie en voûte. Dans leur ensemble de données, trois cas sont possibles : argile molle, argile raide et sable. Plutôt que de fournir des chaînes de caractères incompréhensibles par l'algorithme, ces auteurs ont utilisé trois caractéristiques booléennes pour représenter les données de sol. Cette méthode peut cependant conduire à augmenter considérablement le nombre de caractéristiques et beaucoup de redondance dans les données : on explose un champ unique en plusieurs champs avec beaucoup de valeurs vides.
  Une autre technique couramment utilisée est le label encoding, qui consiste à attribuer à chaque catégorie de la variable un nombre entier unique. Ocak et Seker (2013) et Kim et al. (2022b) ont testé cette méthode pour définir la géologie au front. Par exemple, Kim et al. (2022b) ont utilisé la définition suivante : 1 = remblais, 2 = alluvions, 3 = granite complètement décomposé, 4 = roche altérée en boule (corestone) et 5 = roches. Cependant, le label encoding peut conduire à une mauvaise performance des modèles d'apprentissage automatique car les nombres attribués peuvent donner l'impression de relations d'ordre qui n'existent pas en

réalité. Il est donc important de choisir un encodage approprié au contexte et aux données disponibles.

Cas particulier des paramètres géologiques

Comme expliqué précédemment, la définition correcte d'une stratigraphie nécessite une multitude de paramètres géologiques. La réduction de dimension de ces paramètres spécifiques a été explorée selon plusieurs approches.

- Une des démarches consiste simplement à ne pas inclure la géologie dans les paramètres d'entrée du modèle d'apprentissage automatique, si le creusement se fait dans une stratigraphie plutôt homogène ou peu variable (Boubou et al., 2010). Cette approche est également utilisée pour créer des modèles généralisables pour différentes stratigraphies (Zhou et al., 2023).
- Une autre approche consiste à prendre en compte uniquement le sol au niveau du front du tunnel, en utilisant par exemple le type de géologie au front comme caractéristique. Cette variable catégorielle peut être transformée en variable numérique en utilisant des méthodes comme le one-hot encoding (Kohestani et al., 2017 ; Pourtaghi et Lotfollahi-Yaghin, 2012 ; Suwansawat et Einstein, 2006) ou le label encoding (Kim et al., 2022b ; Mohammadi et al., 2015 ; Ocak et Seker, 2013 ; Zhang et al., 2020a) comme expliqué dans le paragraphe précédent. Une approche similaire d'encodage consiste à considérer une caractéristique booléenne pour la géologie au front (Liu et al., 2022 ; Wang et al., 2013). Cette technique est uniquement applicable dans le cas où seuls deux types de sol sont rencontrés au front, puisque la valeur 0 désignerait un type de sol alors que 1 désigne l'autre type. Une autre méthode propose de prendre en compte le pourcentage des différents types de sol au front. Cette approche est utilisée par Santos et Celestino (2008) qui considèrent deux caractéristiques : le pourcentage d'argile et le pourcentage de sable au front.
- Il est possible de faire une moyenne des paramètres, comme le module élastique et la teneur en eau, pour obtenir des caractéristiques représentatives (Zhang et al., 2021a).
- Une autre approche consiste à prendre en compte uniquement l'épaisseur des couches de la stratigraphie rencontrée par le tunnel. Dans ce cas, le nombre de caractéristiques serait égal au nombre de couches de la stratigraphie (Kim et al., 2022b).
- Chen et al. (2019a) introduisent une autre méthode combinant les paramètres des différentes couches de sol en un jeu unique et qui est utilisée par la suite par Su et al. (2022) et Zhang et al. (2020a). Cette méthode consiste à combiner les propriétés mécaniques des couches de sol, leur épaisseur ainsi que leur position par rapport au

tunnel en utilisant la formule suivante :

$$x = \sum_{i=1}^{n} \frac{e_i}{H} \cdot \frac{h_i}{H} \cdot x_i \tag{3.1}$$

où x est, pour chacune des propriétés, la combinaison des paramètres mécaniques x_i des couches i. Les paramètres géométriques sont définis quant à eux sur la Figure 3.7.

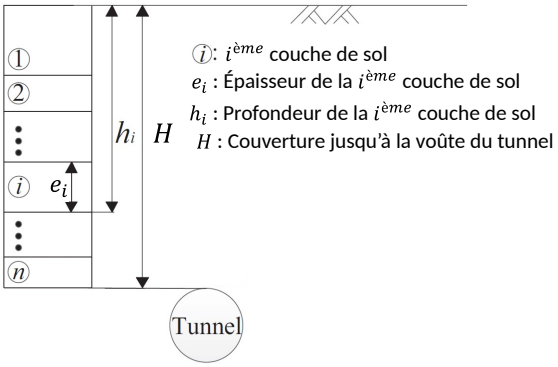


Figure 3.7. Schématisation de la méthode de combinaison des paramètres de sols, adaptée de Chen et al. (2019a)

La présence d'une ou plusieurs nappes phréatiques est un autre facteur qui est parfois pris en compte dans les paramètres d'entrée du sol. Plusieurs approches ont été proposées dans la littérature pour traiter ce facteur. Liu et al. (2022) utilisent le niveau de la nappe comme paramètre d'entrée, tandis que d'autres auteurs ont recours à la distance entre la nappe et la base du tunnel (Kohestani et al., 2017 ; Pourtaghi et Lotfollahi-Yaghin, 2012 ; Santos et Celestino, 2008 ; Suwansawat et Einstein, 2006 ; Wang et al., 2013 ; Zhang et al., 2020a). D'autres études ont simplement utilisé une variable booléenne pour indiquer la présence ou l'absence de la nappe, en fonction de sa distance par rapport à la voûte du tunnel (Dindarloo et Siami-Irdemoosa, 2015 ; Neaupane et Adhikari, 2006 ; Ocak et Seker, 2013).

3.3.3 Mise à l'échelle des caractéristiques

À peu d'exceptions près, les algorithmes d'apprentissage automatique ne fonctionnent pas très bien lorsque les variables numériques en entrée ont des amplitudes de variation très différentes car ces variables n'apportent pas dans ces cas là une contribution égale à l'analyse. Cela est le cas par exemple pour les algorithmes qui se basent sur le calcul de distance (comme les SVM) ou sur le calcul de poids (comme les ANN). De plus, la convergence des algorithmes est plus rapide avec des variables à étendue et variance similaire. C'est la raison pour laquelle la mise à l'échelle des caractéristiques (feature scaling), ou recalibrage, est une technique très intéressante en pratique. On peut noter qu'il n'est en revanche de façon générale pas nécessaire de mettre à l'échelle les valeurs

cibles (Géron, 2022). Il convient de noter que les algorithmes qui se basent sur les arbres de décisions comme DT, RF ou encore XGBoost n'ont pas besoin de normalisation. En effet, les arbres de décision sont construits en divisant progressivement l'espace de recherche en sous-espaces de plus en plus petits, en fonction des caractéristiques des données (§ 2.3.1). Les tests effectués pour diviser l'espace de recherche sont basés sur des seuils qui sont comparés aux valeurs des caractéristiques. Par conséquent, les écarts de valeurs des caractéristiques ne sont pas affectés par les différences d'échelle. Les méthodes de mise à l'échelle les plus communes sont :

– la mise à l'échelle min-max (min-max scaling ou normalization) est la plus simple : les valeurs sont décalées et recalibrées afin qu'elles se situent toutes entre 0 et 1. Pour le faire, il suffit de leur soustraire la valeur minimum, puis de diviser par la valeur maximale – la valeur minimale :

$$\frac{x - min(x)}{max(x) - min(x)}$$

– la normalisation (ou standardisation, standardization) consiste tout d'abord à soustraire la valeur moyenne puis à diviser par l'écart-type. On obtient alors une distribution de moyenne nulle et de variance égale à 1 :

$$\frac{x - moyenne(x)}{écart\ type(x)}$$

Contrairement à la mise à l'échelle min-max, la standardisation ne limite pas les valeurs à un intervalle donné, ce qui pourrait être un problème pour certains algorithmes (les réseaux neuronaux, par exemple, attendent souvent une valeur d'entrée comprise entre 0 et 1). Cependant, la standardisation peut être affectée par les valeurs aberrantes. Il est ainsi recommandé de supprimer les données aberrantes avant de normaliser les données. Santos et Celestino (2008) ont étudié l'influence de la normalisation sur les résultats de prévision des tassements. Cependant, un changement de caractéristiques a également été effectué ce qui ne permet pas de tirer des conclusions claires sur l'influence de la normalisation prise isolément.

3.4 Conception et optimisation des modèles

L'obtention d'un modèle performant nécessite plusieurs étapes. La première étape est l'étape de conception du système. Ensuite, les données sont divisées en sous-ensembles d'entraînement et de test. Enfin, il convient de chercher à optimiser les hyperparamètres en vue d'obtenir du modèle des performances optimales.

Table 3.2. État de l'art sur le choix des variables cibles (tout type de creusement)

Variable cible	Références
s_{max}	Ahangari et al. (2015), Chen et al. (2022), Chen et al. (2019a) Chen et al. (2019b), Dindarloo et Siami-Irdemoosa (2015) Goh et Hefney (2010), Hajihassani et al. (2020) Hasanipanah et al. (2016), Kim et al. (2001) Kohestani et al. (2017), Liu et al. (2022) Mahmoodzadeh et al. (2020), Moeinossadat et al. (2018) Mohammadi et al. (2015), Neaupane et Adhikari (2006) Ocak et Seker (2013), Pourtaghi et Lotfollahi-Yaghin (2012) Qiao et al. (2010), Santos et Celestino (2008) Shi et al. (1998), Su et al. (2022) Suwansawat et Einstein (2006), Zhang et al. (2020a) Zhang et al. (2021a), Zhou et al. (2023)
s^*	Bouayad et Emeriault (2017), Boubou et al. (2010) Kim et al. (2022a), Kim et al. (2022b)
s_{long}	Marto et al. (2012), Wang et al. (2013)

3.4.1 Conception du système

La conception du système consiste à choisir la (les) variable(s) cible(s), les algorithmes d'apprentissage automatique à tester ainsi que les mesures de performance qui permettent de juger de la performance des algorithmes.

Choix de la (des) variable(s) cible(s)

Le tassement induit par le creusement des tunnels est un phénomène tridimensionnel complexe, caractérisé par plusieurs paramètres tels que le tassement maximal le long de l'axe du tunnel (s_{max}, § 1.2.2), le tassement à une distance donnée de l'axe du tunnel (s^*, § 1.2.2) ou encore le tassement à une certaine distance du front du tunnel (s_{long}) (§ 1.2.2). La Table 3.2 synthétise les références étudiées en fonction des variables cibles choisies dans les différentes études.

D'autres auteurs se sont intéressés à la prévision de paramètres supplémentaires tels que la distance au point d'inflexion de la cuvette de tassement dans le sens transversal au creusement (i_y). Par exemple, Neaupane et Adhikari (2006) utilisent deux réseaux de neurones BPNN avec les mêmes caractéristiques. Le premier réseau a été conçu pour prédire le tassement maximal s_{max}, tandis que le deuxième réseau cible la prévision de i_y. Il est important de noter que le deuxième réseau a utilisé s_{max} comme caractéristique supplémentaire. Kim et al. (2001) ont également étudié un exemple similaire, utilisant un seul réseau de neurones BPNN pour prédire s_{max} et i_y. Hajihassani et al. (2020) ont quant à eux prédit les trois paramètres permettant de définir une cuvette tridimensionnelle à l'aide d'un seul PSO-ANN, avec trois variables cibles : s_{max}, i_y et i_x.

Choix des algorithmes

Un choix éclairé des algorithmes d'apprentissage automatique adapté à un problème donnée est un travail primordial si l'on veut obtenir des prévisions de qualité optimale. Lors de la sélection de l'algorithme le plus adapté, plusieurs critères doivent être considérés, tels que la disponibilité des données, le mode d'apprentissage, le mode de collecte des données, la nature de la tâche à effectuer ainsi que les critères d'évaluation du modèle. Il est recommandé de procéder à l'entraînement de plusieurs algorithmes et de comparer leurs performances respectives afin de déterminer celui qui sera le plus approprié pour le cas d'étude considéré. Dans cette optique, il est souhaitable de privilégier au moins en première approche un algorithme facile à utiliser et à interpréter, ce qui facilitera l'analyse des résultats ainsi que la prise de décision ultérieure.

Le mode d'apprentissage supervisé est le mode systématiquement adopté pour la prévision du tassement car la collecte de mesures de surveillance des déplacements du sol permet d'obtenir des données étiquetées. En effet, l'ensemble de données contient les valeurs de sortie (mesures de tassement) en correspondance des données d'entrée (nature du sol et paramètres de creusement). Le processus d'apprentissage supervisé consiste donc à utiliser ces données étiquetées pour entraîner un modèle qui peut ensuite être utilisé pour faire des prévisions sur de nouvelles données.

Il est possible d'ajuster ou de ré-entraîner le modèle à mesure que de nouvelles données sont collectées, ce qui permet ainsi d'améliorer sa performance au fil du temps. Dans le cas présent, cela implique la génération d'un flux de données au fur et à mesure de l'avancement de l'excavation d'un tunnel. Par conséquent, le modèle doit être conçu pour un apprentissage sur flux continu (§ 2.2.2). Cependant, dans un contexte de recherche et développement, les données sont généralement pré-enregistrées et ne présentent pas de flux de données en temps réel. Dans ce cas, l'apprentissage est hors ligne.

La plupart des études considèrent le tassement comme une variable continue et utilisent des modèles de régression pour prédire sa valeur numérique. Toutefois, une alternative consiste à envisager le tassement comme une variable catégorielle, ce qui permet de le traiter comme une tâche de classification. Le choix des catégories pertinentes dépend de la distribution des valeurs de tassement dans l'ensemble de données étudié. Par exemple, Dindarloo et Siami-Irdemoosa (2015) ont opté pour des catégories telles que 0-9.9 mm ; 10-19.9 mm ; 20-20.9 mm ; >30 mm alors que Kim et al. (2022b) ont choisi les catégories $\geq$ 0 mm ; 0-5 mm ; 5-10 mm ; $>$ 10 mm. Le choix entre la régression et la classification dépendra du niveau de précision requis. Si l'on s'intéresse par exemple uniquement à la prévision de deux catégories, "tassement à risque" et "tassement non risqué", des algorithmes de classification pourrait effectivement faciliter le travail.

Dans une étude récente menée par Liu et al. (2022), les méthodes ensemblistes (Ensemble methods, § 2.3.1) telles que les Random Forests (RF) ou les Gradient Boosting Machines (GBM) ont été identifiées comme étant les plus performantes pour la prévision des tassements induits par le creusement des tunnels, et cela même avec de faibles quantités de données. Cette conclusion est soutenue par plusieurs études récentes, notamment

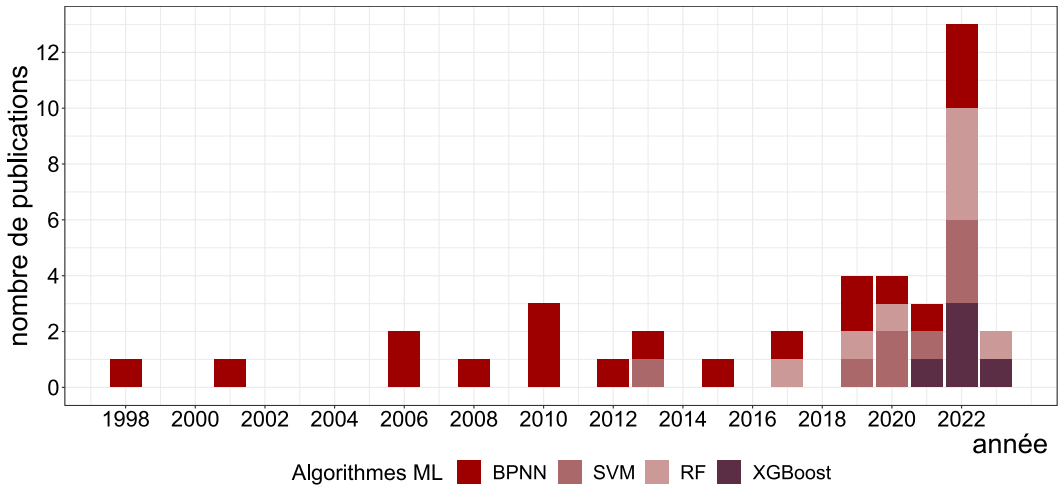


Figure 3.8. Les principaux algorithmes d'apprentissage automatique utilisés dans l'état de l'art pour la prévision des tassements causés par le creusement des tunnels au fil des années

celles menées par Chen et al. (2022), Kim et al. (2022a), Su et al. (2022) et Zhang et al. (2020a). Par conséquent, ces modèles (RF et XGBoost) ont suscité un intérêt considérable ces dernières années en tant que méthodes de prévision fiables pour les tassements (Figure 3.8). En revanche, les réseaux de neurones artificiels (BPNN) demeurent à ce jour la méthode la plus répandue pour la prévision des tassements, suivis par les Support Vector Machines (SVM), Random Forests (RF) et XGBoost (Figure 3.9).

Choix des mesures de performance

Les mesures de performance sont des éléments clés pour évaluer la qualité du modèle. Il convient ainsi de sélectionner des métriques adaptées à la tâche à accomplir (régression ou classification) afin d'apprécier la performance du modèle. Pour une tâche de régression, les métriques couramment utilisées sont la RMSE, la MAE, la MSE et le R^2 (§ 2.3.2, Figure 3.10).

L'incorporation de la physique du problème sous forme de métrique d'erreur représente une étape supplémentaire importante. Dans le cas étudié, la sous-estimation des tassements peut avoir des conséquences préjudiciables. Il est donc important de prendre en compte cette information lors de la sélection des métriques de performance. Ainsi, l'erreur quantile (Équation 2.7) peut être utilisée pour pénaliser les sous-estimations des tassements par le modèle, favorisant ainsi des prévisions plus sûres. Un exemple concret est fourni par Liu et al. (2022), qui ont utilisé un coefficient γ de 0,7 pour l'erreur quantile.

Il convient aussi de mesurer la stabilité du modèle. Un modèle d'apprentissage automatique est considéré comme stable lorsqu'il est capable de produire des résultats cohérents et précis sur un large éventail de données d'entrée. Plus précisément, un modèle stable est capable de généraliser correctement à partir des données d'entraînement et

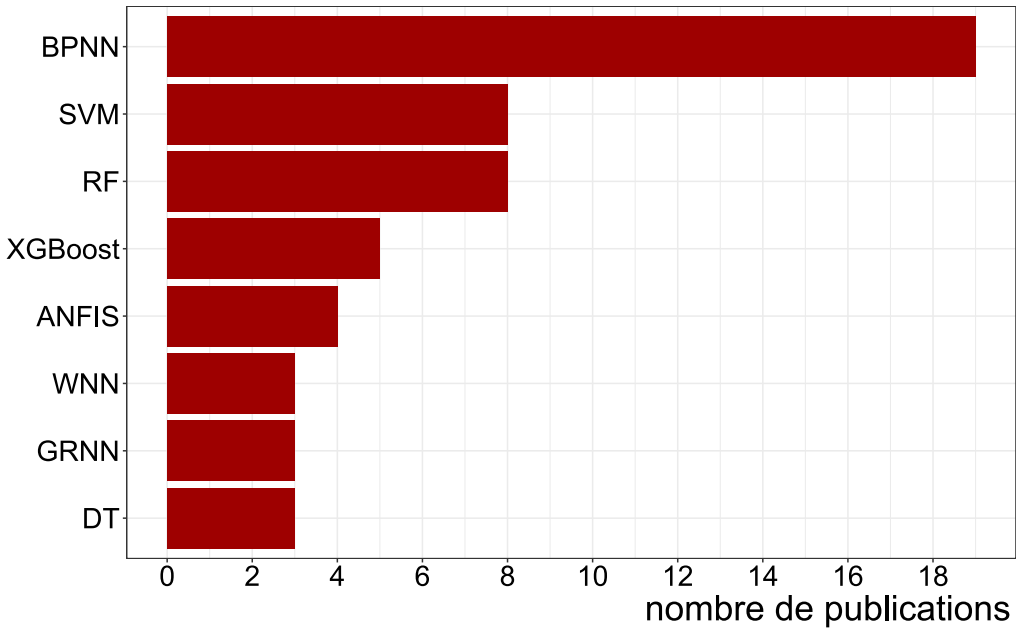


Figure 3.9. Les principaux algorithmes d'apprentissage automatique utilisés dans l'état de l'art pour la prévision des tassements causés par le creusement des tunnels.
Légende : BPNN, SVM, RF, XGBoost, ANFIS, WNN, GRNN, DT

d'obtenir des performances similaires sur les données de test. Un modèle instable, en revanche, peut produire des résultats extrêmement variables en fonction des données d'entrée, ce qui le rend imprévisible et peu fiable. Cette instabilité peut être due à un sur-apprentissage ou un sous-apprentissage. Par conséquent, pour garantir la stabilité d'un modèle, il est important d'utiliser des techniques telles que la régularisation, la validation croisée et l'optimisation des hyperparamètres. Il convient également de vérifier a posteriori qu'il n'y a pas eu de sur ou sous-apprentissage. De plus, il est important de disposer d'un ensemble de données d'entraînement de qualité et suffisamment représentatif pour garantir que le modèle est capable de généraliser correctement aux données réelles. Liu et al. (2022) utilisent la variance s^2 pour évaluer la stabilité du modèle. La variance est donnée par la formule suivante :

$$s^2 = \frac{\sum(X - \overline{X})}{n - 1}$$

où X sont les prévisions, $\overline{X}$ la moyenne des prévisions et n le nombre de répétition des prévisions. Plus les données sont dispersées, plus la variance par rapport à la moyenne est importante. Les résultats de cette étude débouche sur un classement de la stabilité des modèles testés (du plus stable au moins stable) : RF > SVM > GBM > DT > FPI-BPNN > BPNN.

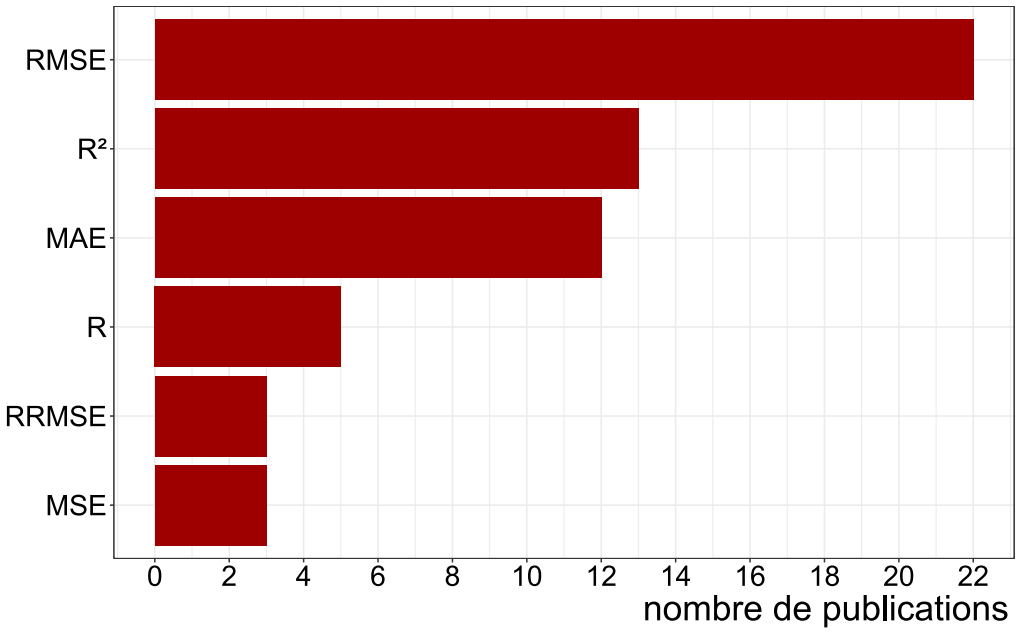


Figure 3.10. Les principales mesures de performance utilisés dans l'état de l'art pour la prévision des tassements.
Légende : RMSE, R^2, MAE, R, MSE, RRMSE

3.4.2 Division des données

Lors de la division des données pour l'entraînement (train) et l'évaluation (test) d'un modèle d'apprentissage automatique, il faut déterminer si le but final est l'interpolation (ou généralisation) ou l'extrapolation. Avant de discuter de cette question, il convient de clarifier les différences entre ces deux concepts. L'interpolation consiste à prédire des valeurs pour des données qui se trouvent à l'intérieur de la plage de données connues du modèle. Autrement dit, un modèle qui interpole n'est capable que de prédire des valeurs pour des entrées qui se situent à l'intérieur des limites des données d'entraînement. En revanche, un modèle qui extrapole est capable de prédire des cibles pour des entrées qui se situent en dehors de la plage de données connues par le modèle. Il est couramment admis que les algorithmes d'apprentissage automatique sont conçus essentiellement pour l'interpolation et ne peuvent pas garantir des performances satisfaisantes en extrapolation (Ye, 2020). Toutefois, certains modèles peuvent être entraînés à mieux extrapoler en utilisant des techniques telles que la régularisation des algorithmes (en optimisant les hyperparamètres), l'augmentation de données et l'utilisation de données synthétiques.

Pour réaliser des interpolations, les données sont généralement divisées de manière aléatoire en ensembles d'entraînement et de test, afin de disposer d'un nombre suffisant de données pour entraîner le modèle et évaluer sa capacité de généralisation. Il convient de noter que l'augmentation de la taille de l'ensemble de données d'entraînement améliore les performances du modèle, mais peut également augmenter le temps de calcul et le risque de sur-apprentissage (overfitting). Cependant, si la proportion de données utilisées pour l'entraînement est trop faible, le modèle ne sera pas en mesure de reproduire ou de prédire

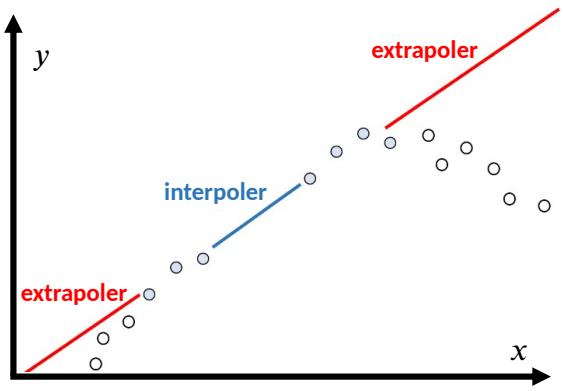


Figure 3.11. Différence entre l'interpolation et l'extrapolation : un algorithme entraîné sur les données en gris extrapole sur les données en dehors de la plage initiale (en blanc). Adaptée de Bobbitt (2021)

correctement les sorties souhaitées. La manière courante de déterminer la proportion de données à utiliser pour l'entraînement est de réaliser une analyse d'optimisation (Boubou et al., 2010). En général, environ 80% des données sont réservées à l'entraînement, tandis que le reste est utilisé pour le test (Figure 3.12). Cependant, certains auteurs, comme Wang et al. (2013), ont choisi de n'utiliser que 20% des données pour l'entraînement (soit 147 mesures de tassement), en utilisant les 514 autres mesures pour le test. Les résultats montrent des valeurs de RMSE de l'ordre de 4 mm.

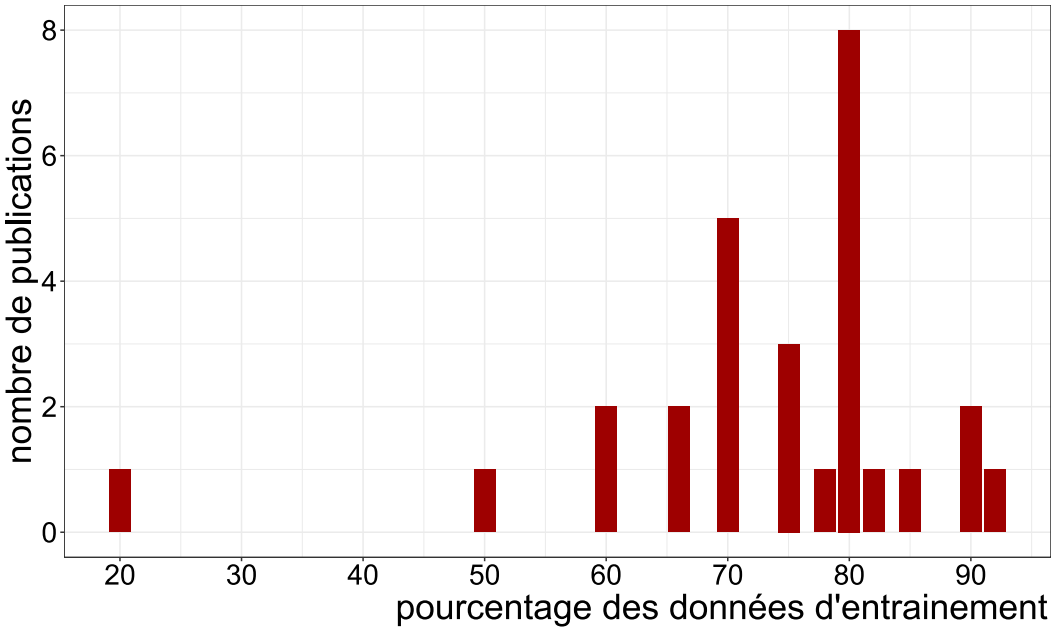


Figure 3.12. Répartition du pourcentage de données pris pour l'entraînement des algorithmes d'apprentissage automatique pour la prévision des tassements induits par le creusement des tunnels

Pour obtenir des interpolations précises, il est important de s'assurer que la plage de données de test est similaire à celle des données d'entraînement. Les distributions des données peuvent être visualisées pour confirmer que les ensembles de données sont

comparables et représentatifs de toutes les situations pour lesquelles le modèle sera utilisé (Figure 3.13).

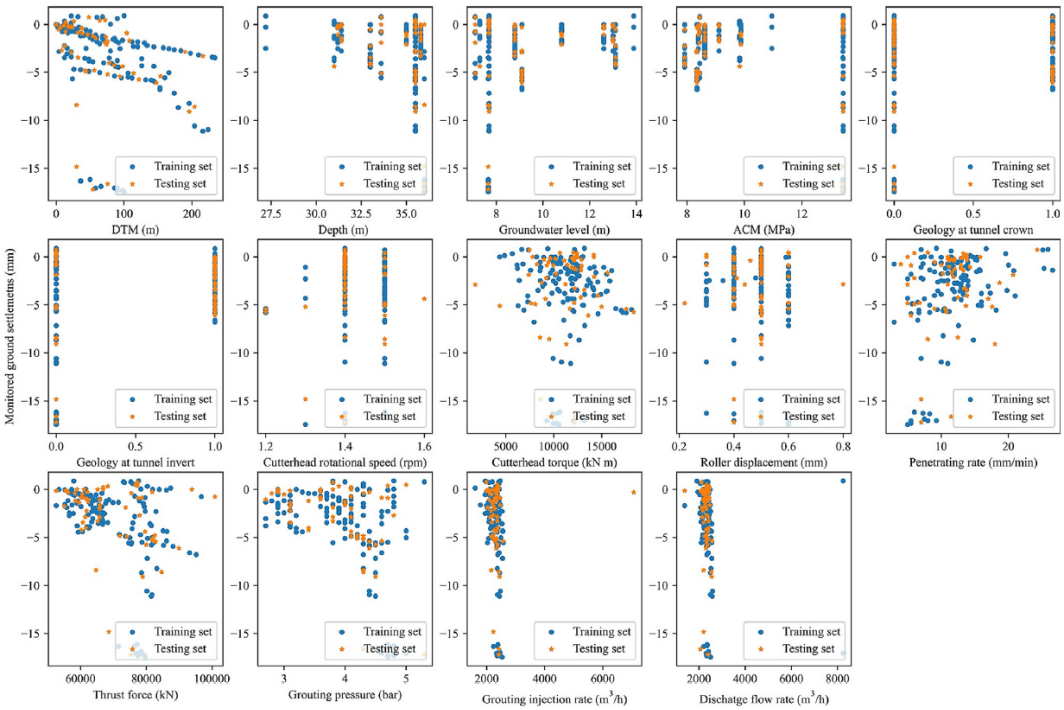


Figure 3.13. Distribution des données d'entraînement et de test (Liu et al., 2022)

D'autres méthodes de division des données sont possibles afin de tester les capacités d'extrapolation du modèle. Par exemple, Suwansawat et Einstein (2006) disposent d'un ensemble de données de 49 observations obtenues de 4 sections de tunnels creusés au tunnelier à pression de terre. Ces auteurs considèrent la division de données selon les trois scénarios suivants :

1. 70% des données d'une section sont pris aléatoirement pour l'entraînement et les 30% restant de cette même section sont utilisés pour le test.
2. Les premiers 50% des données d'une section sont pris pour l'entraînement et les derniers 50% de la même section sont pris pour le test.
3. 70% des données des 4 sections sont pris aléatoirement et le reste des 30% sont utilisés pour le test.

Les résultats montrent que le dernier scénario a les résultats les moins intéressants. Cela pourrait être dû au fait que les tunneliers utilisés pour le creusement des différentes sections sont fournis par des fabriquants différents (Herrenknecht et Kawasaki). Par conséquent, une caractéristique supplémentaire est ajoutée pour prendre en compte la machine de creusement, ce qui a légèrement amélioré les résultats. Il convient de noter que la taille des ensembles de données utilisés est beaucoup trop petit pour pouvoir entraîner correctement un réseau de neurones (environ 30 mesures au total pour chaque section).

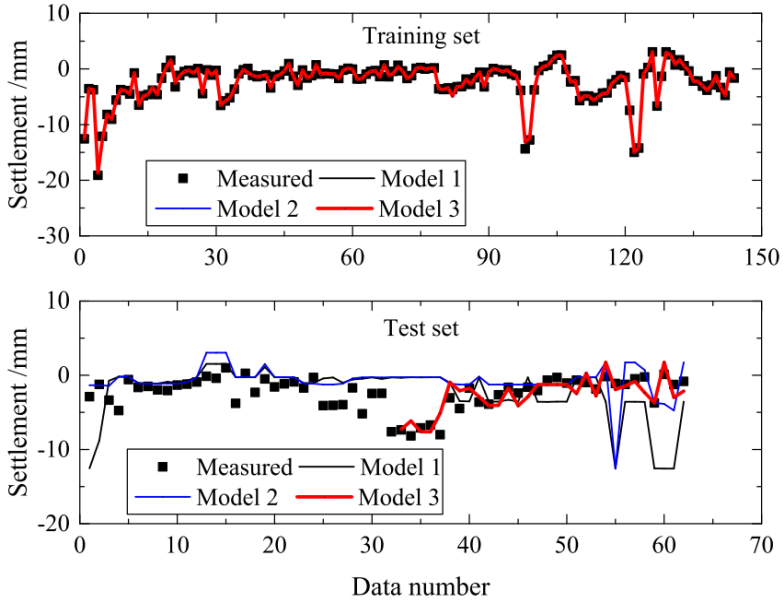


Figure 3.14. Comparaison des résultats des trois scénarios proposés par Chen et al. (2019a)

Un autre exemple est effectué par Chen et al., 2019a. Ces auteurs avaient également accès à 4 sections de tunnels creusés au tunnelier et ont considéré la division des données selon trois scénarios :

1. entraînement sur la première section (39 mesures) et test sur la quatrième section (62 mesures),
2. entraînement sur les trois premières sections (138 mesures) et test sur la quatrième section,
3. entraînement sur les trois premières sections et la moitié de la quatrième section (169 mesures) et test sur le reste de la 4ème section (31 mesures).

Les résultats obtenus sont présentés dans la Figure 3.14. On en déduit que le modèle est plus performant avec le 3ème scénario ce qui pousse à conclure qu'il est important de faire des mises à jour des modèles (ré-entraînement) tout au long du creusement du tunnel.

Boubou et al. (2010) ont également essayé de prédire le tassement transversal s^* en entraînant leur modèle sur le début du creusement de la ligne puis en l'évaluant sur le reste de la ligne. Ces auteurs ont varié la longueur de la ligne utilisée pour l'entraînement afin de trouver le nombre de données qui donne les meilleures performances. Les résultats obtenus montrent qu'un entraînement sur 1500 m (245 mesures) et un test sur 400 m (76 points) donne les meilleurs résultats (Figure 3.15).

A noter que dans les trois articles de Boubou et al. (2010), Chen et al. (2019a) et Suwansawat et Einstein (2006), un modèle initial est utilisé afin de trouver l'architecture optimale vis-à-vis des hyperparamètres et c'est le modèle optimal qui est utilisé dans tous les scénarios.

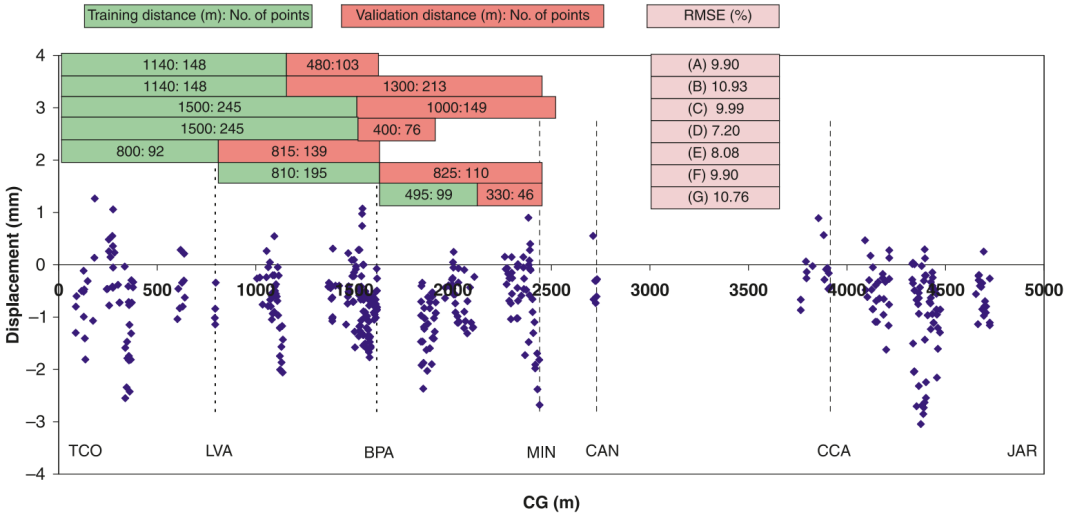


Figure 3.15. Résultats des études de Boubou et al. (2010).
Légende : TCO, LVA, BPA, MIN, CAN, CCA, JAR représentent les gares.

3.4.3 Optimisation des hyperparamètres

L'optimisation des hyperparamètres est un processus crucial dans la construction de modèles d'apprentissage automatique, car les hyperparamètres peuvent avoir un impact significatif sur les performances du modèle (§ 2.2.1). Les méthodes traditionnelles pour optimiser les hyperparamètres, telles que la validation croisée et la recherche exhaustive de l'espace de recherche (« grid search »), peuvent être coûteuses en temps et en ressources, surtout lorsque l'espace de recherche est grand. La recherche exhaustive est testée dans l'étude de Kim et al. (2022b) entre autres.

L'optimisation bayésienne (Bayesian optimization) est une méthode d'optimisation des hyperparamètres plus rapide et efficace, en particulier pour les modèles complexes, qui utilise des modèles probabilistes. Cette méthode utilise des connaissances préalables sur l'espace de recherche pour guider la recherche vers les régions les plus prometteuses de l'espace de recherche. L'optimisation bayésienne a été utilisée avec succès dans les études de Kim et al. (2022a) et Su et al. (2022).

Particle Swarm Optimization est une autre méthode d'optimisation populaire pour les problèmes d'optimisation non linéaires. Cette méthode simule le comportement d'un groupe de particules pour trouver la meilleure solution possible. Cette méthode est utilisée par Zhang et al. (2020a), Hasanipanah et al. (2016) et Hajihassani et al. (2020) pour optimiser les hyperparamètres de modèles de prévision de tassement.

Enfin, de nouvelles méthodes d'optimisation continuent d'être proposées, telles que le Seagull Optimization Algorithm, introduit par Dhiman et Kumar (2019). Cette méthode simule le vol d'un groupe de mouettes pour optimiser une fonction objectif. La méthode a été testée avec succès par Zhou et al. (2023).

Conclusion

Ce chapitre était dédié à la description de l'état de l'art de la prévision des tassements à l'aide de méthodes d'apprentissage automatique, tout en exposant la méthodologie de travail. Une brève introduction sur l'état de l'art de l'apprentissage automatique appliqué à la géotechnique en général a également été présentée. La méthodologie de travail est décomposée en trois parties distinctes.

Dans un premier temps, une analyse exploratoire des données est essentielle pour la compréhension des données. Cette étape comprend l'exploration des données, leur nettoyage, leur stockage et des analyses statistiques. Ces dernières permettent l'extraction d'informations utiles, comme la détection de relations potentielles entre les variables.

La seconde étape consiste en l'établissement d'une ingénierie des caractéristiques. Cette tâche requiert trois étapes. La première consiste à sélectionner les variables pertinentes pour la prévision de la variable cible, en se basant sur les connaissances métier et les analyses statistiques. La seconde consiste en l'extraction des caractéristiques à introduire dans l'algorithme d'apprentissage automatique. Cela peut inclure des combinaisons de variables, des transformations, ou encore une réduction de dimensions. Le cas particulier de l'extraction de caractéristiques représentant le sol est détaillé. Enfin, la troisième étape consiste à mettre à l'échelle les caractéristiques, étape qui peut être nécessaire selon les algorithmes d'apprentissage sélectionnés à l'étape suivante.

La dernière étape de la méthodologie consiste à concevoir le système, c'est-à-dire sélectionner la variable cible, les algorithmes d'apprentissage automatique ainsi que les mesures de performance. Quelques exemples de division des données pour l'entraînement et l'évaluation sont décrits, ainsi que l'optimisation des hyperparamètres pour obtenir les meilleures performances.

En résumé, ce chapitre présente la méthodologie de travail de cette thèse, en s'appuyant sur des exemples de l'état de l'art de la prévision des tassements par des algorithmes d'apprentissage automatique. Les études précédentes ont révélé que les forêts aléatoires RF et les XGBoost sont les modèles qui ont donné les meilleures performances, et que les mesures de performances les plus utilisées sont le RMSE, R^2 et MAE. Cependant, certaines limitations ont été identifiées, telles que la taille réduite des ensembles de données utilisés et le manque de prise en compte de la progression du tassement avec l'avancement, qui n'a été abordée que par Wang et al. (2013). Il convient de souligner qu'une analyse approfondie de l'état de l'art permet de constater l'émergence d'une certaine maturation scientifique sur les sujets relatifs à l'apprentissage automatique, offrant des perspectives prometteuses pour de futures applications.

La deuxième partie de ce manuscrit est dédiée à l'application exhaustive de cette méthodologie sur un large ensemble de données recueillies lors du creusement de sections des lignes 14 sud et 15 sud-ouest du projet du Grand Paris Express. L'objectif principal est de proposer une preuve de concept, POC (preuve de concept) sous forme d'un algorithme robuste et fiable, capable d'être déployé dans le cadre de la construction de nouvelles lignes de métro afin de prédire avec précision les tassements. En particulier, par la suite,

on visera à prendre en compte tout d'abord l'aspect spatial puis l'aspect spatio-temporel qui sont très peu développés à ce jour.

Conclusion

La revue de l'état de l'art présentée dans cette partie fournit des connaissances fondamentales nécessaires à la réalisation de cette thèse.

En premier lieu, nous avons acquis une compréhension approfondie de la description précise des tassements en surface, résultant de la construction de tunnels, à travers les équations transversales et de progression du tassement en fonction du creusement du tunnel (Chapitre 1).

Ensuite, nous avons approfondi les techniques d'apprentissage automatique afin de les appliquer dans la suite avec une bonne compréhension de leur fonctionnement (Chapitre 2).

Enfin, nous avons également examiné la méthodologie de travail utilisée pour cette thèse à travers l'étude de l'état de l'art de la prévision des tassements à l'aide d'algorithmes d'apprentissage automatique (Chapitre 3). Cette méthodologie comprend une analyse exploratoire des données pour extraire des relations potentielles entre les variables, l'ingénierie des caractéristiques pour sélectionner les variables pertinentes et extraire les caractéristiques à utiliser dans l'algorithme d'apprentissage automatique avant l'application de ces algorithmes. De plus, ce chapitre nous a permis de voir qu'aujourd'hui, la majorité des études utilisent une division aléatoire des données, ne prenant pas en compte la réalité spatio-temporelle du creusement et de la progression des tassements. Cela constituera un point important des travaux de cette thèse (Partie III).

La deuxième partie de ce manuscrit est dédiée à l'ingénierie des données du Grand Paris Express : construction de la base de données ainsi que l'analyse exploratoire et l'ingénierie des caractéristiques.

Partie II

Ingénierie des données du Grand Paris Express

Introduction

Cette partie est dédiée à l'étude approfondie des données obtenues à partir de deux lignes de métro du projet du Grand Paris Express, soit la ligne 14 Sud et la ligne 15 Sud-Ouest. Ces études commencent d'abord par la collecte des données, leur nettoyage et organisation dans une base de données. Ensuite, sont effectuées des extractions de données supplémentaires à partir des mesures, telles que les paramètres des équations de tassement. Finalement, des analyses statistiques sont menées pour détecter les tendances dans les données ainsi que les relations entre les différents paramètres de creusement ou de sol et les tassements observés en surface.

Il a été jugé approprié de rassembler toutes ces études sous le nom d'« ingénierie des données », étant donné que l'objectif est de tirer parti des informations disponibles pour les utiliser ultérieurement dans des simulations basées sur l'apprentissage automatique. Il convient de commencer par une description générale du projet du Grand Paris Express.

Description du projet du Grand Paris Express

Le Grand Paris Express (GPE) est le plus grand projet urbain d'Europe avec à terme la construction de plus de 200 km de lignes automatisées de métro. Ce projet global d'aménagement de la métropole répond à la forte croissance urbaine en Île-de-France. Dans le cadre de cette thèse, nous exploitons les données de creusement d'une partie des lignes 14 Sud (L14S) et 15 sud-ouest (L15SO) du GPE. Avant d'aborder le traitement et l'exploitation de ces données, on présente ici le projet du GPE, la géologie et les méthodes de creusement des L14S et L15SO.

Contexte Général du Grand Paris Express

Le projet du GPE est la réalisation de quatre nouvelles lignes de métro autour de la capitale (15, 16, 17 et 18) et le prolongement de la ligne 14 au nord, de Saint-Ouen à Saint-Denis, et au sud, entre Paris et l'aéroport d'Orly (Figure 3.1). Avec 200 km de lignes, soit presque autant que le linéaire de métro actuel, et 68 gares, ce futur réseau de transport desservira les proche et grande couronnes. Il sera mis progressivement en service entre 2024 et l'horizon 2030.

La conception et la réalisation de ce projet ont été actées par la loi n° 2010-597 du 3 juin 2010 qui définit le GPE comme « un projet urbain, social et économique d'intérêt national » qui vise à promouvoir « le développement économique durable, solidaire et créateur d'emplois de la région capitale » (Art. 1). Le maître d'ouvrage de ce projet est la Société du Grand Paris (SGP) qui est un établissement public chargé d'organiser la conception et la réalisation du GPE. Ce projet représente à date un investissement de 36.1 milliards d'euros, financé par des recettes fiscales franciliennes, le recours à l'emprunt auprès d'investisseurs publics et des levées de fonds sur les marchés financiers dans le

cadre de programmes labellisés « green ». Le projet bénéficie également de subventions européennes (SociétéGrandParis, 2022).

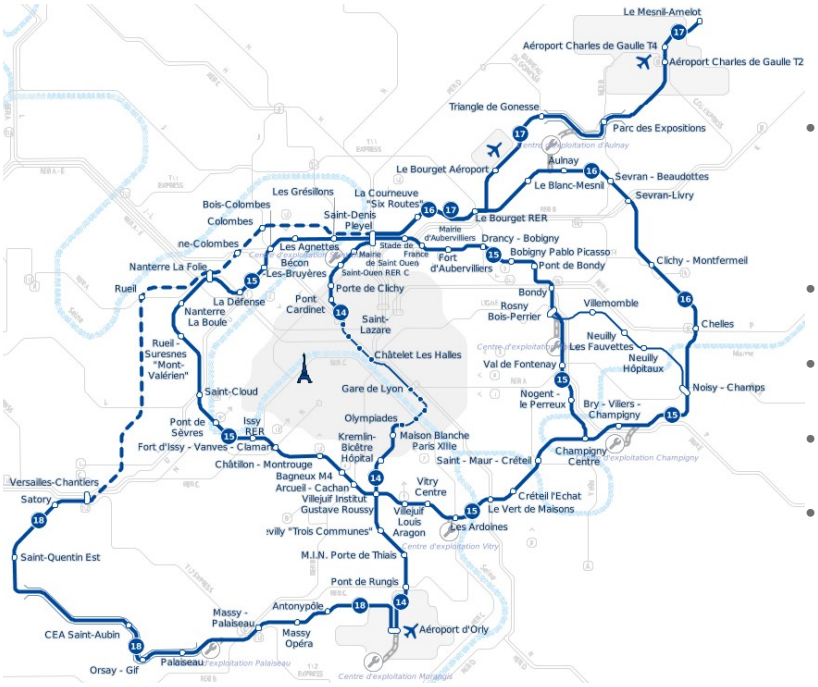


Figure 3.1. Carte Globale et chiffres clés du projet du Grand Paris Express (adaptée de SociétéGrandParis (2022))

Ligne 14 Sud du Grand Paris Express

La ligne 14 Sud (L14S) constitue la prolongation de la ligne 14 du métro parisien entre la station Olympiades et l'aéroport d'Orly. Dans le cadre de cette thèse, nous avons uniquement accès aux données du lot GC02 de la L14S, soit 4 578 m. Ce lot comprend la réalisation du tunnel qui est un ouvrage monotube à deux voies, de diamètre intérieur 7,75 m, foré au tunnelier entre l'ouvrage « Jean Prouvé » au Sud et la gare « Maison Blanche – Paris $XIII^{ème}$ » au Nord. Le lot GC02 inclut également la réalisation des ouvrages annexes « Jules Guesde », « Marcel Sembat », « Cuchets », « République » et « Jean Prouvé » (Figure 3.2), lequel sert de puits de départ pour le tunnelier. Ces ouvrages sont situés dans les départements de Paris (75) et du Val-de-Marne (94), et sur les communes suivantes : Paris $XIII^{ème}$, Le Kremlin Bicêtre, Villejuif, Arcueil, L'Haÿ-Les-Roses. Dans ce qui suit, la notation L14S2 désigne spécifiquement le lot GC02.

Ligne 15 Sud-Ouest du Grand Paris Express

La ligne 15 Sud-Ouest (L15SO) s'étend sur une longueur de 12 km de l'ouvrage annexe « Ile de Monsieur » (ouvrage inclus) à la gare de « Villejuif Louis Aragon » (gare incluse). Entièrement en souterrain, elle traverse principalement des zones urbaines denses et

Figure 3.2. Gares et Ouvrages Annexes des zones étudiées des lignes 14 Sud (à gauche) et 15 Sud-Ouest (à droite)

intègre trois passages sous-fluviaux. La L15SO dessert 8 gares en correspondance avec des transports urbains et ferroviaires existants ou en projet. Par la suite, on étudie le tronçon T3C de la L15SO qui a pour objet la réalisation d'un tunnel foré de 7785 m et de 8.7 m de diamètre intérieur entre les gares de « Villejuif Louis Aragon » (gare incluse) et la gare de « Fort d'Issy-Vanves-Clamart » (gare partiellement incluse), de 8 ouvrages annexes (y compris le rameaux de raccordement entre le puits et le tunnel foré) (Figure 3.2). Le tunnel se décompose en deux secteurs creusés par 2 tunneliers :

– Le premier secteur (Centre, également noté TR1) s'étend du tympan Est de la gare de « Fort d'Issy-Vanves-Clamart » jusqu'au tympan Ouest de l'ouvrage annexe « OA-P04 – Parc Robespierre » sur une longueur de 3 800 m environ. Le tunnelier du TR1 est monté et lancé dans l'ouvrage « OA-P04 – Parc Robespierre » et assure le forage du tunnel jusqu'à la gare de « Fort d'Issy-Vanves-Clamart » où il est démonté.

– Le deuxième secteur (Est, également noté TR2) s'étend du tympan Est de l'ouvrage annexe « OA-P04 – Parc Robespierre » jusqu'au tympan Ouest de la gare de « Villejuif Louis Aragon » sur une longueur de 3985 m environ. Le tunnelier du TR2 est monté et lancé dans le puits de départ de la gare d'« Arcueil Cachan » jusqu'à la gare « Villejuif Louis Aragon » où il est démonté.

– Le tronçon entre l'ouvrage « OA-P04 – Parc Robespierre » et la gare d'« Arcueil Cachan » (noté TR3) est creusé par le tunnelier du TR1.

La majeure partie du tracé se situe sous des bâtiments résidentiels et industriels dont certains sont sensibles. Le tracé passe également sous l'autoroute A6, sous d'anciennes carrières de Calcaires Grossier ainsi que sous des voies ferrées de la SNCF et de la RATP.

CONSTRUCTION D'UNE BASE DE DONNÉES

4

Introduction

Le projet du Grand Paris Express (GPE), notamment les lignes 14 Sud (L14S2) et 15 Sud-Ouest (L15SO), génère une grande quantité de données. Cependant, sans traitement préalable, ces données restent inexploitables. Dans le cadre de cette étude, nous avons collecté, organisé, nettoyé et stocké les données dans une base de données. Cette tâche a nécessité un investissement très conséquent, car il était essentiel de réfléchir à un concept et une architecture convenables pour la future utilisation de la base de données. En effet, cette dernière contient des informations sur la nature des sols rencontrés, les paramètres de pilotage du tunnelier ainsi que sur les mesures de tassements observées en surface.

Le présent chapitre vise à présenter un panorama général des lignes L14S2 et L15SO, ainsi que des géologies rencontrées, avant de détailler la procédure de traitement des données.
Bien que la construction de la base de données ne soit pas l'objectif principal de cette thèse, elle demeure d'une importance considérable. En effet, cette base de données offre une valeur ajoutée significative à cette étude ainsi qu'à de futurs travaux sur ces données. La structure de cette base de donnée pourrait également servir à accueillir les données de futurs projets du même type. Elle est donc susceptible de vivre en dehors du cadre strict de ce travail de thèse pour permettre de déployer à des échelles supérieures les outils conçus et évalués dans ce travail de recherche. La présentation de cette procédure de traitement des données est donc essentielle pour comprendre les résultats et les conclusions de cette thèse, ainsi que pour faciliter l'utilisation ultérieure des données.

4.1 Description des lignes étudiées

Dans cette étude, les données proviennent de la ligne 14 Sud (L14S2) et des tronçons TR1, et TR2 et TR3 de la ligne 15 Sud-Ouest (L15SO). Cette partie détaille le contexte géologique rencontré en plus de quelques informations utiles sur le creusement de ces lignes.

4.1.1 Panorama général

Les deux lignes 14 Sud et 15 Sud-Ouest sont excavées par des tunneliers à pression de terre manufacturés par Herrenknecht (Figure 1.2). Les étapes de creusement se déroulent de la manière suivante. Tout d'abord, la machine excave une longueur égale à celle d'un anneau de voussoirs avant de s'arrêter. Une fois arrêté, l'érecteur pose les 7 voussoirs

qui forment l'anneau. Pour les lignes étudiées, les anneaux ont une longueur de 2 m, à l'exception de quelques anneaux de la L14S2 qui font 1.5 m. Les vérins peuvent ainsi s'appuyer sur le nouvel anneau pour permettre au tunnelier de continuer le creusement pour poser l'anneau suivant. En parallèle du creusement, du mortier de bourrage est injecté à l'arrière de la jupe pour combler le vide annulaire. Si N est le dernier anneau posé, l'injection de mortier se fait alors à l'arrière de l'anneau N-1 lors du creusement de l'anneau N+1. Pour une jupe de 10 m et des anneaux de 2 m, l'injection se fait donc 14 m à l'arrière de la roue de coupe.
Une série de photos prises au cours du creusement d'un tronçon de la Ligne 15 Sud-Ouest du projet du Grand Paris Express est proposée dans la Figure 1.6. Une description plus détaillée du mode de fonctionnement des tunneliers à pression de terre est présentée dans le § 1.1.2.

Les dates de creusement des tronçons sont les suivantes :

- Le tronçon TR1 de la L15SO a été creusé entre le 28/02/2019 et le 07/08/2020 ; 1954 anneaux ont été posés.
- Le tronçon TR2 de la L15SO a été creusé entre le 29/04/2019 et le 11/12/2020 ; 1645 anneaux ont été posés.
- Le tronçon TR3 de la L15SO a été creusé entre le 14/12/2020 et le 06/04/2021 ; 341 anneaux ont été posés.
- Le tronçon L14S2 de la L14S a été creusé entre le 05/08/2019 et le 14/03/2021 ; 2316 anneaux ont été posés.

Des statistiques sur le nombre d'anneaux posés par jour sont présentées dans le § 5.3.1.

4.1.2 Contexte géologique

Les formations géologiques rencontrées au front le long du tracé de la L14S2 sont des Argiles Plastiques et le Calcaire Grossier entre les gares de Maison Blanche et du Kremlin-Bicêtre puis les Marnes et Caillasses , le Marno-Calcaire de Saint-Ouen et les Masses et Marnes du Gypse jusqu'à la gare de Villejuif Institut-Gustave-Roussy et enfin les Argiles Vertes et des Marnes Supra-Gypseuses jusqu'à la fin du tronçon.

En ce qui concerne les conditions hydrogéologiques pour la L14S2, deux nappes superposées impactent le tracé dans sa première partie, celle du Calcaire Grossier (mur constitué par l'Argile Plastique sous-jacente) puis celle de l'Yprésien Sableux situé sous l'Argile Plastique. Dans la deuxième partie du tracé, les nappes interceptées sont celle des Marnes de Pantin portée par les Argiles Vertes, des nappes localisées dans les bancs calcaires des Masses et Marnes du Gypse et celle, profonde, du Marno-Calcaire de Saint-Ouen portée par les Sables de Beauchamp sous-jacents.

Pour la L15SO, le profil en long entre les gares de Fort d'Issy-Vanves-Clamart et Arcueil Cachan est majoritairement dans les Argiles Plastiques, les Marnes de Meudon et le Calcaire Grossier. Le profil traverse par la suite, à l'est d'Arcueil Cachan, la vallée de la

Bièvre qui est le point le plus critique par rapport aux effets induits par les tassements en surface. A ce point, le tunnel sort du Calcaire Grossier jusqu'à effleurer les Éboulis sur la pente du Plateau de Villejuif. Entre Villejuif Institut-Gustave-Roussy et la fin du tracé, les formations rencontrées sont ensuite de bas en haut les Marnes et Caillasses, les Sables de Beauchamps, le Calcaire de Saint-Ouen, les Marnes Infra-Gypseuses, les Masses et Marnes du Gypse et les Marnes Supra-Gypseuses.

Les terrains de la L15SO sont baignés par plusieurs nappes superposées en raison de l'hétérogénéité de nature des couches et des variations locales de topographie. Entre Issy-les-Moulineaux et Arcueil se trouvent trois nappes : une nappe superficielle dans le Marno-Calcaire de Saint-Ouen résiduels soutenue par les Sables de Beauchamps, une nappe principale dans le Calcaire Grossier, soutenue par l'Argile Plastique et une nappe profonde sous l'Argile Plastique, dans les Marnes de Meudon et la Craie. La vallée de la Bièvre est quant à elle concernée par la nappe alluviale de la rivière Bièvre. Enfin, le plateau de Villejuif est commun avec la L14S2 déjà décrit ci-avant.

Les profils en long des deux lignes sont illustrés dans la Figure 4.1. On peut constater que l'épaisseur de la couverture au-dessus des tunnels varie, selon la ligne, entre 15 m et 50 m. Les principales formations rencontrées au front des deux lignes sont présentées dans la Figure 4.2.

4.2 Traitement des données

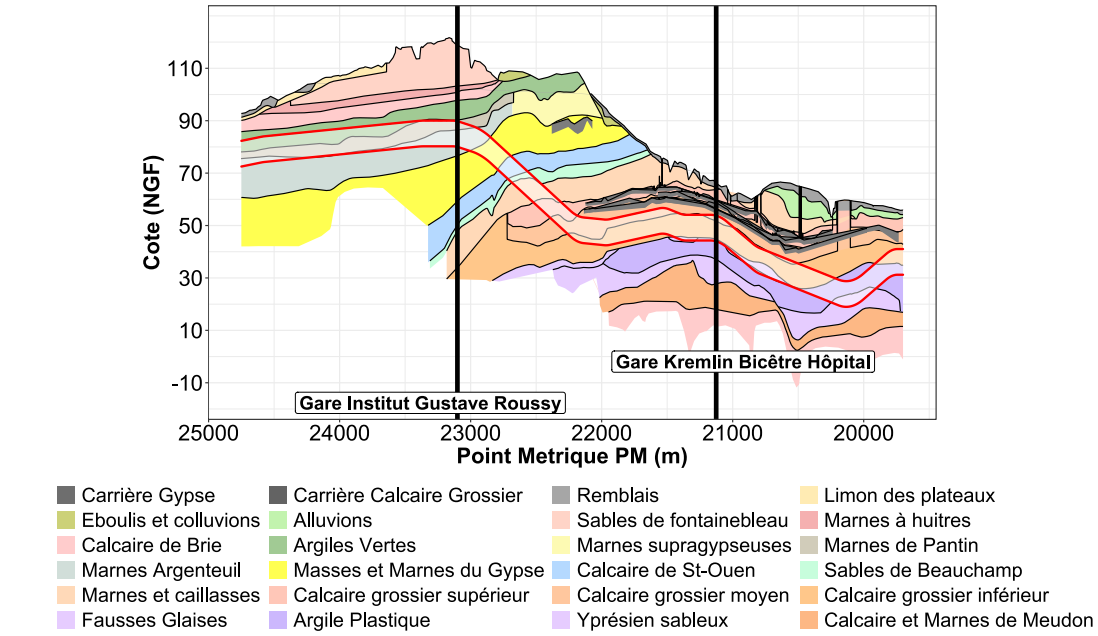
Le traitement des données (data processing) englobe l'ensemble des étapes allant de l'extraction des données jusqu'à leur stockage sous forme d'informations exploitables (Talend, 2023).

4.2.1 Extraction des données

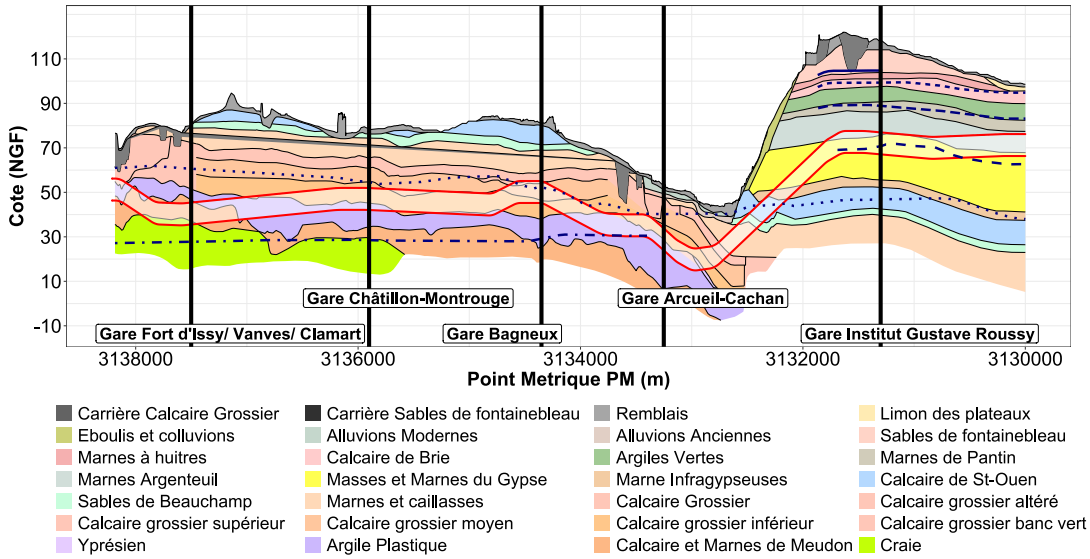
Comme présenté dans le Chapitre 3, les paramètres d'entrée peuvent être divisés en trois catégories : la géométrie du tunnel, les conditions géologiques et les paramètres de creusement, auxquels s'ajoute la variable cible qui est la valeur du tassement mesuré en surface. Cette section présente ces données dans le cadre du GPE ainsi que les techniques d'extraction de données utilisées.

Contexte général

La plupart des données relatives au GPE sont mises à disposition sur diverses plateformes pendant les travaux de construction. Préalablement à toute étude sur ces données, un travail d'extraction des informations brutes est nécessaire. Le premier défi est l'acquisition des données dans les délais puisqu'une fois l'exécution des travaux achevée, l'accès aux plateformes mettant les données à disposition des acteurs impliqués n'est plus possible.



(a) L14S2



(b) L15SO

Figure 4.1. Profil en long des lignes L14S2 et L15SO (Richa et al., soumis)

Il est donc impératif d'extraire les informations et d'en faire une sauvegarde. Le second défi est la diversité des plateformes fournissant les paramètres nécessaires à nos études. L'accès à ces plateformes se fait par différentes méthodes (application, site web, API), et l'extraction de masse n'est souvent pas évidente (outils non adaptés). Le défi à l'heure actuelle est donc de faire avec, de s'adapter à ces disparités, et d'arriver à créer des outils permettant cette exportation de façon fiable et automatisée.

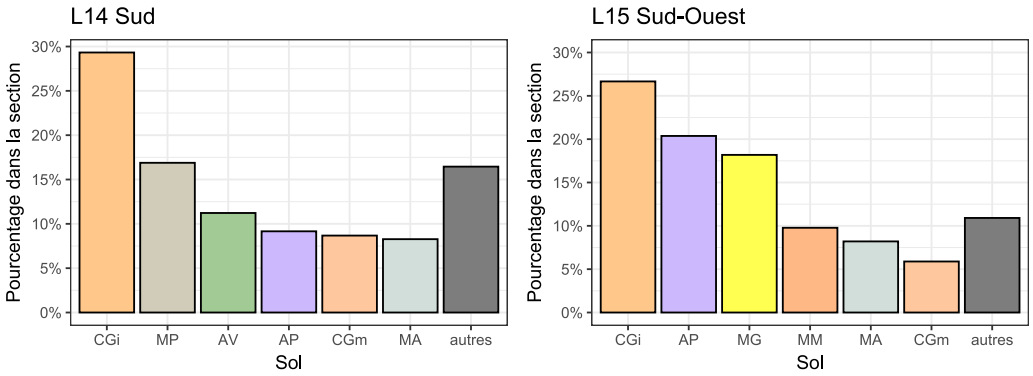


Figure 4.2. Principales formations rencontrées au front des lignes 14 Sud et 15 Sud-Ouest (Richa et al., 2022).
Légende : CGi, CGm, AV, AP, MA, MP, MG, MM

Conditions géométriques

Dans cette catégorie de données se trouve majoritairement la couverture de sol au-dessus du tunnel et le diamètre de creusement. Ce dernier est naturellement uniquement pris en compte lorsque l'ensemble des données contient plusieurs diamètres de creusement. Ces deux paramètres sont généralement définis depuis le début du projet et sont faciles à obtenir depuis les cartes numériques ou des plans numérisés.

D'autres paramètres peuvent s'ajouter à cette catégorie comme la distance du front du tunnel à la station de départ, la distance à l'axe du tunnel des cibles de mesures de tassement posées en surface ou encore la distance entre une cible et le front du tunnel (d_{front}). L'obtention de ces paramètres nécessite de passer par les plateformes qui donnent les données de pilotage de tunnelier ainsi que les données relatives aux cibles et leurs mesures.

Un paramètre supplémentaire est la position sur le tracé, représentée par un point kilométrique PK qui est une position absolue sur le linéaire d'un tunnel, à partir d'un point de référence souvent variable dans la vie d'un projet, et exprimé en kilomètres. On utilise également la notion de point métrique PM lorsque cette référence est exprimée en mètre. Le tracé étant connu dans l'espace, l'usage est donc d'exprimer une position de la machine (roue de coupe par exemple), par son PK, noté pk_{rdc}. Il convient de noter que le creusement ne se fait pas toujours dans le sens croissant des PK.

Conditions géologiques

Avant le creusement du tunnel, les conditions géologiques sont préalablement établies afin de dimensionner le tunnel en fonction de la qualité des sols traversés. Les caractéristiques mécaniques du sol qui reflètent son comportement et sa nature sont déterminées par une multitude de paramètres. Ces derniers, qui sont les paramètres interprétés par les ingénieurs géotechniciens, sont obtenus à partir de données brutes acquises par la réalisa-

Table 4.1. Définition des secteurs. L'ordre est celui du creusement des tunnels.

id_secteur	Tronçon	Nom	Longueur
6	TR1	plateau de Malakoff à Bagneux hors gares Bagneux et Châtillon-Montrouge	450
5	TR1	autour de la gare Bagneux	250
4	TR1	plateau de Malakoff à Bagneux hors gares Bagneux et Châtillon-Montrouge	1300
3	TR1	autour de la gare Châtillon-Montrouge	400
2	TR1	plateau de Malakoff à Bagneux hors gares Bagneux et Châtillon-Montrouge	1200
1	TR1	autour de la gare Fort d'Issy/ Vanves/ Clamart	950
10	TR2	plateau de Villejuif hors gare Intitut-Gustave-Roussy	1300
9	TR2	autour de la gare Intitut-Gustave-Roussy	700
8	TR2	vallée de la Bièvre y compris coteau de Villejuif	1250
7	TR2	autour de la gare Arcueil-Cachan	650
6	TR3	plateau de Malakoff à Bagneux hors gares Bagneux et Châtillon-Montrouge	450
21	L14SGC02	Ouvrage Annexe Jean Prouvé	100
20	L14SGC02	Ouvrage Annexe République	100
19	L14SGC02	Intergare Villejuif Intitut-Gustave-Roussy - Chevilly 3 Communes	1450
18	L14SGC02	Gare Intitut-Gustave-Roussy	200
17	L14SGC02	Ouvrage Annexe Cuchets	100
16	L14SGC02	Ouvrage Annexe Marcel Sembat	200
15	L14SGC02	Intergare Kremlin Bicêtre - Villejuif IGR	1675
14	L14SGC02	Gare Kremlin Bicêtre	300
13	L14SGC02	Ouvrage Annexe Jules Guesde	100
12	L14SGC02	Intergare Maison Blanche - Kremlin Bicêtre	900
11	L14SGC02	Intergare Maison Blanche - Kremlin Bicêtre	525

tion d'essais in-situ ou en laboratoire. Ils sont consignés dans des rapports de synthèse géologique et géotechnique (« cahiers B » pour les projets de tunnel selon la nomenclature AFTES) par plages de linéaires, qu'on appelle généralement secteurs. La liste des secteurs est donnée dans la Table 4.1. Dans le cadre de cette thèse, nous avons donc choisi de partir exclusivement des données déjà interprétées et non des données brutes issues des essais, dont le traitement, le nettoyage et l'ordonnancement aurait constitué une tâche complexe et chronophage. Nous avons par ailleurs, pour ce travail de thèse, circonscrit le panel de paramètres interprétés disponibles à quelques paramètres jugés a priori clés vis-à-vis du problème posé d'évaluation des tassements. Ce choix a été effectué avec les géotechniciens ayant travaillé sur ces projets. Les paramètres retenus sont donc les suivants :

- le poids volumique en place γ [kN/m^3]
- le module pressiométrique Ménard E_M [MPa]
- le coefficient rhéologique α
- la cohésion c [kPa]
- l'angle de frottement φ [°]

– le coefficient de pression des terres au repos K_0.

Ces paramètres sont fournis pour chaque couche de sol et chaque secteur dans des tables dispersées qu'il convient de consolider en une table unique. La table source des paramètres interprétés prend donc la forme après ce travail manuel d'une table à double entrée : paramètres en fonction de la couche de sol (***Parametre_Mecanique_Formation***).
La stratigraphie peut-être extraite mètre par mètre à partir des profils en long dessinés par les ingénieurs Ce travail d'extraction à partir des plans permet de produire une table source (***Stratigraphie***) dont les champs sont les suivants : PK, cote du terrain naturel z_{TN} [m], une colonne pour chaque couche de sol possible avec comme valeurs la cote de la base de la couche ou bien NA en cas d'absence de cette couche.

Pilotage du tunnelier

Les paramètres relatifs au pilotage du tunnelier sont affichés en temps réel sur les écrans de la cabine de pilotage de la machine (Figure 1.6e). L'entreprise n'a pas toujours mis à disposition de la maîtrise d'oeuvre pour ces projets l'ensemble des flux de données brutes (données numériques sous forme de tables exploitables). Néanmoins, elle a mis à disposition des captures d'écran régulières des informations visibles dans la cabine de pilotage, prises toutes les 30 secondes et déposées sur une plateforme web dédiée. Par conséquent, afin d'accéder aux paramètres manquants dans les données brutes, utiles à cette étude, il a été nécessaire d'exploiter également ces captures d'écran (Figure 4.3).

Dans ce contexte, des scripts, dont nous fournissons quelques extraits par la suite, ici en langage R, ont été développés pour extraire automatiquement et massivement des informations ciblées dans toutes ces images (Script 4.1). Des techniques de traitement d'image ont permis au préalable d'améliorer la résolution des caractères avant de les transformer en texte grâce à des librairies de reconnaissances de caractères OCR (Script 4.2 et Figure 4.4).

```
# {R}
# Creation de la session:
session <- rvest::html_session("https://***/login.shtml")
# Lien specifique vers l'image:
link_cap <- "***/screen=Rapports%20Anneau/Rapport%20Anneau%200001.jpg"
# Informations de connexion:
form <- rvest::html_form(session)[[1]] %>%
    rvest::set_values(w_Login = "****", w_PassWord ="****")
rvest::submit_form(session, form)
# Ouverture du lien:
session <- rvest::jump_to(session, link_cap)
# Extraction du contenu de la page:
img_dwl <- session$response$content
# Lecture de l'image:
rapport <- magick::image_read(img_dwl)
# Sauvegarde de l'image:
```

```
magick::image_write(image = rapport)
```

Script 4.1 Extraction d'images depuis le web

```
# {R}
# Caracteres a identifier dans l'image:
digits <- tesseract::tesseract(options = list(
  tessedit_char_whitelist = "-.0123456789"))
digits_date <- tesseract::tesseract(options = list(
  tessedit_char_whitelist = "/0123456789"))
# Permet d'eviter les erreurs telles que la confusion entre le chiffre 5
    et la lettre S ou le chiffre 0 et la lettre O
# zone de texte dans l'image (en pixel):
image_text <- magick::image_crop(rapport, "53x13 + 206 + 40")
# Amelioration de la qualite du texte dans l'image:
image_text <- image_text %>%
  magick::image_resize("180") %>%
  magick::image_enhance() %>%
  magick::image_threshold(type = "white", threshold = "70%") %>%
  magick::image_threshold(type = "black", threshold = "60%")
# OCR:
text <- tesseract::ocr(image_text, engine = digits) %>%
  as.numeric()
text_date <- tesseract::ocr(image_text, engine = digits_date) %>%
  as.Date(format = "%d/%m/%Y")
```

Script 4.2 Extraction des caractères depuis des images (OCR)

Il est à noter que les informations recherchées se trouvent dans différents types d'images, tels que des rapports pour l'injection du mortier ou pour l'avancement du front, et que l'emplacement du texte, de type chiffre ou date, varie selon le rapport. Ainsi, il a été nécessaire de rechercher au cas par cas les pixels associés au texte demandé, ce qui a rendu ce travail assez chronophage. La réalisation de ces tâches pourrait très probablement être améliorée elle-même à l'aide d'algorithmes d'apprentissage automatique, mais nous avons choisi de focaliser nos recherches sur la seule problématique posée.

Les paramètres sélectionnés pour cette étude ont été choisis en se basant sur les paramètres les plus utilisés dans l'état de l'art (Figure 3.6) à savoir :

- la vitesse d'avancement $V_{tunnelier}$ [mm/min]
- le moment de la roue de coupe M_{RDC} [kN.m]
- la pression au front P_{front} [bar]
- la pression d'injection du mortier $P_{mortier}$ [bar]
- la quantité de mortier injecté $V_{mortier}$ [m^3]
- la poussée totale du tunnelier P_{totale} [kN]
- le guidage horizontal G_H [mm] et vertical G_V [mm] du tunnelier

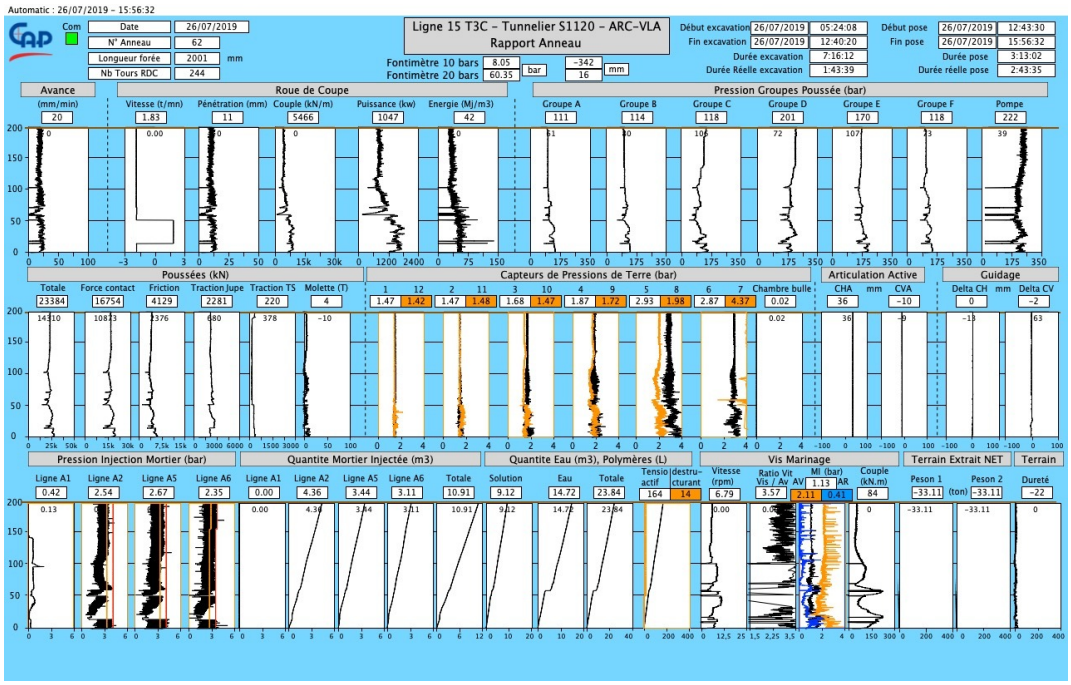


Figure 4.3. Exemple d'image de rapport présentant les paramètres de pilotage du tunnelier

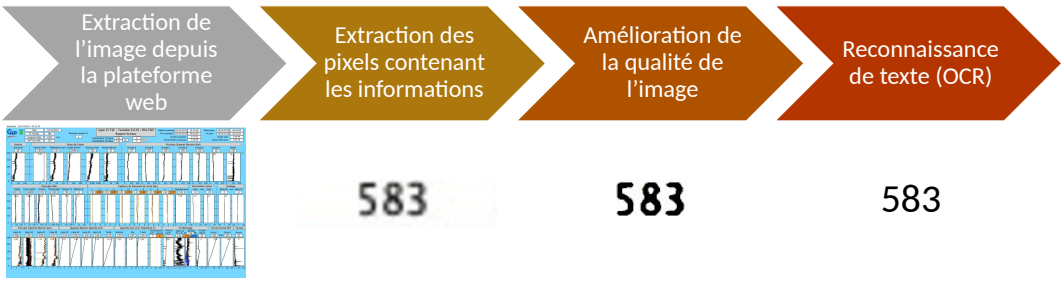


Figure 4.4. Méthode d'extraction des paramètres de pilotage du tunnelier à partir d'images (Richa et al., 2022)

Mesures de tassement

Pour assurer le suivi des déformations causées par le creusement des lignes de métro du GPE, un système d'auscultation automatisé a été mis en place pendant toute la durée des travaux. Ce système utilise une série de théodolites automatiques qui enregistrent les déplacements du sol et des bâtiments avec une fréquence de mesure élevée. Les mires de surface sont visées par les théodolites toutes les 30 minutes environ pour mesurer les déplacements dans les trois directions de l'espace. Environ 16 000 capteurs de différents types ont été placés en voirie ou sur des bâtiments, de part et d'autre de l'axe du tunnel, afin de couvrir la Zone d'Influence Géotechnique (ZIG) (Figure 4.5). Parmi ces points de mesure on compte des points de type « centaures », sans mire physiquement implantée, pour lesquels les théodolites mesurent le déplacement en un point donné, procédant par balayage d'une maille virtuelle. Il est d'usage de considérer que ces mesures sans mire sont moins précises, plus bruitées que celles avec cible physique. Il convient de noter que des mesures d'interférométrie sont également disponibles pour le creusement de ces deux

lignes de métro, mais celles-ci ne seront pas étudiées dans le cadre de cette thèse car cela aurait requis des traitements spécifiques et une gestion de l'hétérogénéité de la précision intrinsèque des mesures au niveau des algorithmes d'apprentissage.

Les mesures enregistrées par les théodolites sont stockées en temps réel dans une base de données, qui peut être consultée à distance par les différents acteurs du projet (entreprise, maître d'œuvre, maître d'ouvrage) via un accès sécurisé à travers un site web ou une application sous forme de plateforme interactive (application Geoscope de l'entreprise Sixsense pour les projets qui nous intéressent). Étant donné que l'exportation massive des données de cette base n'a pas possible via l'application, nous avons du recourir à une technique de type web scraping pour extraire plus de 144 millions de mesures associées à près de 16 000 capteurs de tous types. Le script utilisé pour cette extraction est présenté dans l'Annexe A. Cette extraction a permis d'obtenir deux tables pour chaque tronçon, soit un total de 134 tables pour les deux lignes L15SO et L14S2 (Figure 4.6). La première table contient les propriétés des capteurs, y compris leur nom, leurs coordonnées, leur type, etc. La deuxième table contient les valeurs des mesures tassement avec en colonne les noms des différents capteurs et en ligne les dates des mesures.

Les nombreuses tables obtenues suite à l'extraction des données sont consolidées en quatre tables finales (deux pour chaque ligne de métro) :

– Propriétés : 8 493 lignes pour la L14S2 et 7 366 lignes pour la L15SO.
  Les colonnes sont : tronçon, id, capteur (nom du capteur), x (initial), y (initial), type.
– Mesures : 91 460 640 lignes pour la L14S2 et 52 870 912 lignes pour la L15SO.
  Les colonnes sont : tronçon, date, capteur, mesure (de tassement).

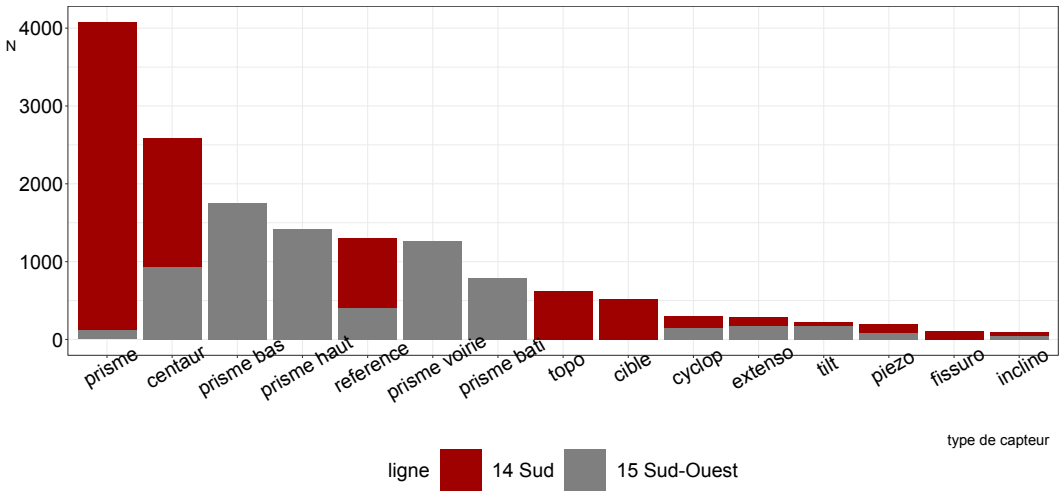


Figure 4.5. Répartition des capteurs selon le type et la ligne de métro

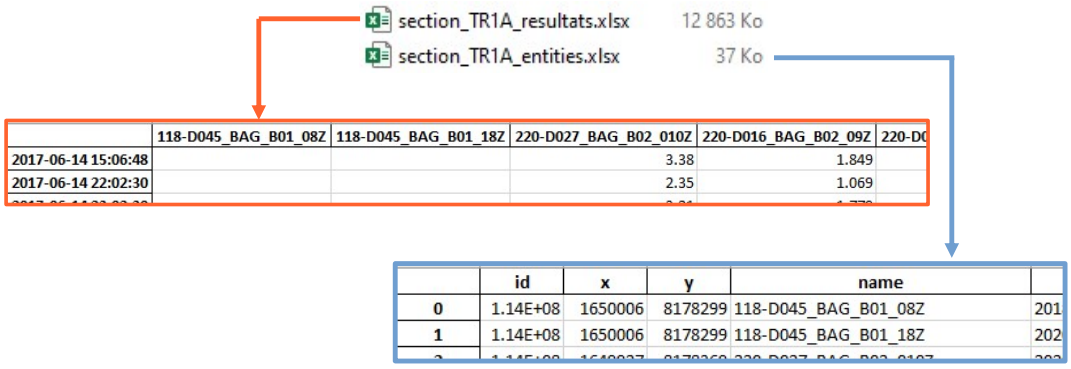


Figure 4.6. Exemple de sortie de l'extraction des données de mesures de tassement

4.2.2 Organisation et Nettoyage

Le nettoyage des données consiste tout d'abord à les organiser dans un format propre et ordonné (tidy data) pour ensuite les nettoyer. Le nettoyage primaire peut inclure le traitement des données manquantes, dupliquées, aberrantes (outliers) ou bruitées ainsi que la jointure des tables et le filtrage de données si besoin. Il convient de noter que le travail est plus itératif que linéaire, c'est-à-dire que les étapes décrites ne se succèdent pas nécessairement mais plutôt se font ensemble et se complètent.

Organisation des données

L'organisation des données est une étape essentielle pour obtenir des données propres et bien rangées (tidy data). Pour obtenir des tables considérées comme propres, il faut respecter les conditions suivantes (Wickham, 2014) :

- chaque variable doit avoir sa propre colonne
- chaque observation doit avoir sa propre ligne
- chaque valeur doit avoir sa propre cellule

Par exemple, la transformation de la table ***Stratigraphie*** (§ 4.2.1) permet de dépivoter les colonnes pour obtenir des données au format long, organisées sous la forme suivante : PK, z_{TN} [m], formation, h (cote de la base de la couche de sol). De même, la table ***Parametre_Mecanique_Formation*** (§ 4.2.1) peut être transformée pour obtenir des données propres selon la forme suivante : secteur, formation et le reste des paramètres mécaniques tels que γ [kN/m^3], E_M [MPa], α, c [kPa], φ [°], K_0. La transformation de ces tables permet notamment d'utiliser des librairies graphiques qui facilitent grandement la programmation nécessaire pour obtenir des profils longitudinaux et des coupes, tels que présentés dans la Figure 4.1b. En tout état de cause, ce format est le seul compatible avec une utilisation de ces données dans le cadre d'une base de données.

Gestion des données manquantes

La gestion des valeurs vides est un problème incontournable lors du traitement de données. Une solution courante consiste à remplir cette donnée manquante par la moyenne des valeurs qui l'entourent. Cependant, cette approche n'est pas toujours applicable.

Dans le cas des données de sol, plusieurs carrières sont rencontrées le long du tracé du tunnel (Figure 4.1a), dont certaines sont remblayées sans que les paramètres mécaniques associés ne soient connus. Dans ce cas, on utilise les paramètres mécaniques du remblai le plus proche. A l'inverse, le remplissage des carrières souterraines de calcaire grossier, souvent de faible épaisseur, a été ignoré et on a choisi de lui associer les paramètres mécaniques des calcaires encaissant. Ce choix est issu de retours d'expériences de calculs de tassements tunnels aux éléments finis en présence de carrières souterraines de faible épaisseur. Outre les carrières, il existe plusieurs couches de sol pour lesquelles il n'y a pas de paramètres mécaniques associés dans des secteurs spécifiques. Pour combler ces lacunes, on utilise les paramètres des mêmes couches dans les secteurs les plus proches.

En ce qui concerne les données de pilotage du tunnelier récupérées, on constate parfois l'absence de certains rapports ou des manques de données pour certains paramètres. Il convient de noter qu'il a été vérifié que ce manque n'est pas lié au travail d'extraction de texte précédemment décrit. Pour les paramètres tels que la vitesse d'avancement et la pression au front, il est possible de recourir à une interpolation pour remplir les lacunes. Toutefois, pour les dates manquantes, il est nécessaire de consulter un autre type de rapport qui présente les dates de début et de fin d'excavation sous forme d'une échelle colorée. Les dates ont alors été corrigées de façon semi-automatisée ce qui est un travail fastidieux mais nécessaire.

En ce qui concerne les mesures de tassement, les capteurs n'ayant pas de coordonnées sont supprimés, car il est impossible de connaître leur position par rapport à l'axe du tunnel. C'est le cas d'un seul capteur pour la L14S2. De plus, il existe des capteurs défectueux de différents types dont les propriétés sont données mais qui n'ont aucune valeur mesurée (Figure 4.7). Ces capteurs sont systématiquement supprimés, soit 685 capteurs au total pour les deux lignes. Ils sont représentés sur la carte de la Figure 4.8. On remarque que la distribution des capteurs est « homogène », ce qui veut dire qu'il n'y a pas de problème spécifique.

Suppression des données dupliquées

Les données dupliquées sont les enregistrements multiples qui se trouvent dans un ensemble de données. Cela peut se produire pour diverses raisons, telles que des erreurs de saisie de données, des erreurs de traitement, des fusions de données incorrectes ou des problèmes de synchronisation de données entre différents systèmes. Les données dupliquées peuvent causer des problèmes lors de l'analyse des données, car elles peuvent fausser les résultats et provoquer des erreurs dans les calculs statistiques. De plus, elles peuvent prendre de l'espace de stockage supplémentaire inutilement.

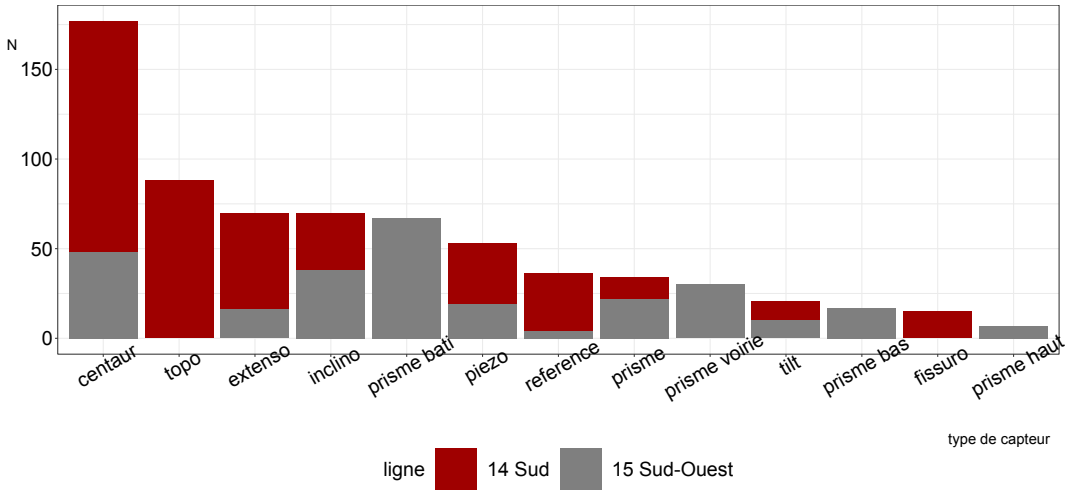


Figure 4.7. Répartition des capteurs sans mesures enregistrées selon leurs types et la ligne de métro

Dans notre ensemble de données, les données dupliquées sont majoritairement observées pour les données relatives aux capteurs et à leurs mesures. Le premier cas de doublons peut être causé par la présence de caractères spéciaux tels que les accents, qui doivent être éliminés. Ainsi, l'élimination des doublons de propriétés de capteurs permet de supprimer 119 et 35 des entrées doublées de capteurs des L14S2 et L15SO, respectivement.

Le deuxième cas de doublons peut résulter des mesures qui ne sont pas des duplications au sens strict mais des données trop proches qui doivent être traitées comme telles pour que les données soient considérées comme propres. C'est le cas par exemple des mesures avec un grand nombre de décimales enregistrées. Pour remédier à cette situation, les déplacements en mm sont arrondis à trois décimales, puis les doublons sont supprimés, réduisant ainsi la taille des tables. Par conséquent, environ 44 millions et 12 millions de mesures en doublon sont supprimées des mesures des lignes L14S2 et L15SO, respectivement.

Détection des données aberrantes

Les données aberrantes, également appelées valeurs atypiques ou outliers en anglais, sont des observations qui diffèrent considérablement du reste des données d'un échantillon. Elles peuvent être causées par des erreurs de mesure, des erreurs de saisie, ou encore par des phénomènes réels mais rares ou extrêmes. L'identification et la gestion des données aberrantes sont un enjeu majeur dans l'analyse de données, car leur présence peut influencer significativement les résultats des analyses.

Ainsi, il est important de mettre en place des méthodes pour détecter et traiter ces valeurs aberrantes. La visualisation graphique des paramètres en fonction de la date est une des méthodes les plus évidentes pour donner à l'analyste la capacité de détecter les données aberrantes et de remonter éventuellement à la source de ce qui les a produites.

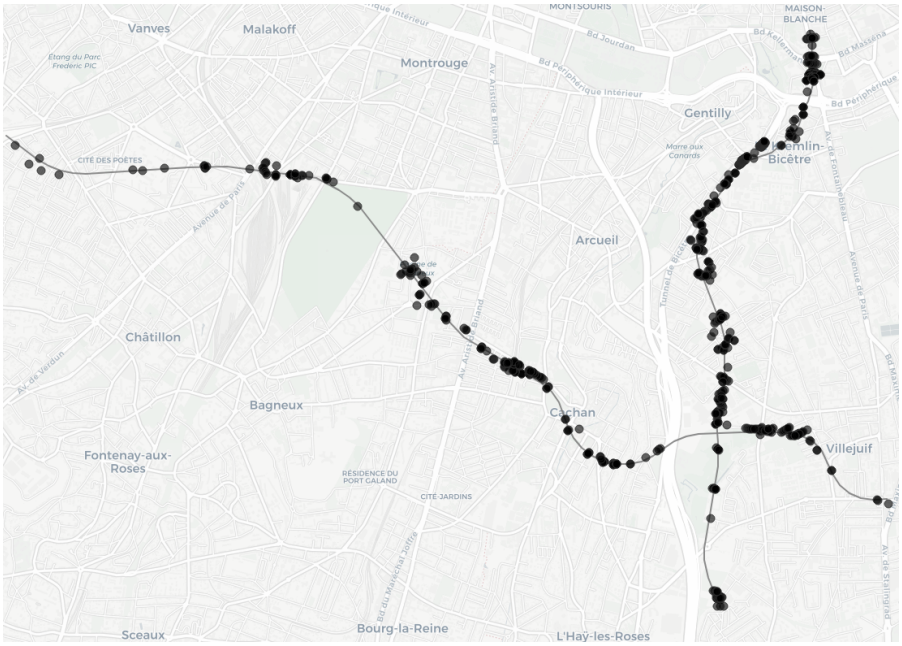


Figure 4.8. Tracés des deux lignes 14 Sud et 15 Sud-Ouest avec les cibles n'ayant pas de mesures enregistrées

Par exemple, on observe dans la Figure 4.9 deux valeurs atypiques de numéro d'anneau. C'est un problème spécifique à la détection de caractère à partir d'images (OCR) où le chiffre 8 est détecté au lieu du chiffre 5.

En ce qui concerne les mesures de tassements, plusieurs méthodes sont appliquées et seront présentées dans le § 5.2.1.

4.2.3 Stockage des données

La dernière étape du traitement des données consiste à les stocker dans une base de données. Dans cette partie, nous allons exposer l'intérêt, l'architecture ainsi que la mise en œuvre de cette base de données de creusement au tunnelier.

Intérêt d'une base de données

Suite à la collecte et au nettoyage des données, on se retrouve avec des fichiers de différents types (csv, json, etc.) sans relation entre eux, stockés dans différents dossiers et trop volumineux pour être chargés en mémoire pour leur exploitation. Travailler avec des données stockées de cette façon n'est pas possible à notre échelle. Nous avons donc décidé de créer une base de données relationnelle pour stocker l'ensemble de ces informations (§ 2.1.2). Bien que la quantité de données à notre disposition ne soit pas encore à l'échelle du Big Data, l'intégration des informations disponibles dans une base de données et le respect des 5V (Volume, Vitesse, Variété, Valeur et Véracité) est un prérequis indispensable pour assurer l'efficacité de la chaîne de valeur.

La mise à disposition d'une base unique garantit à des utilisateurs spécifiques une

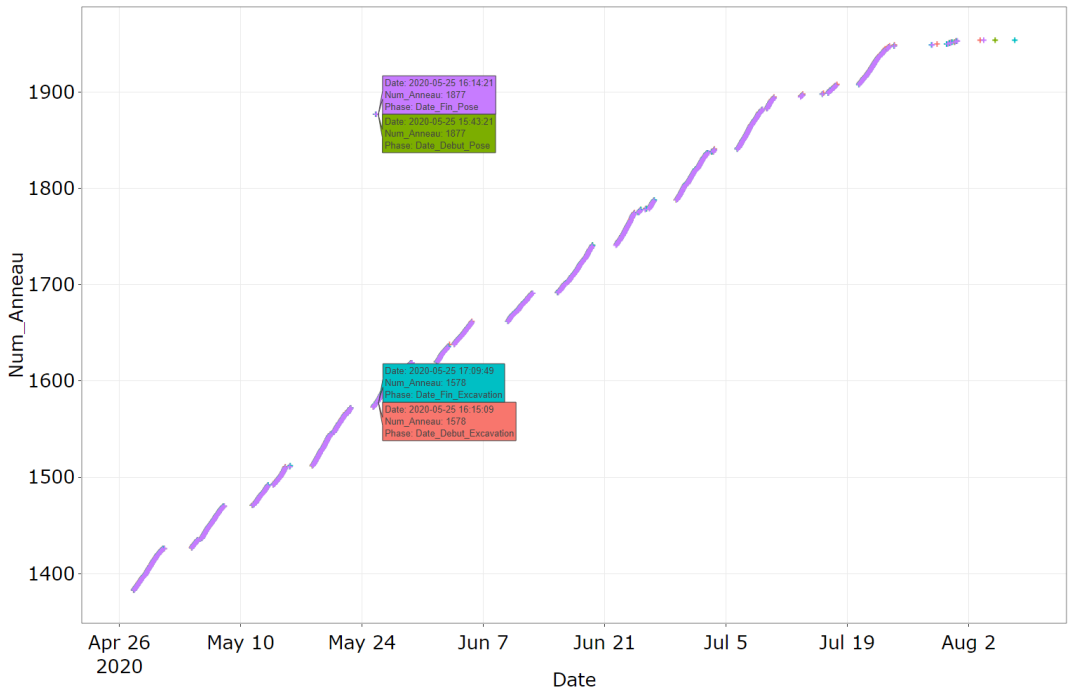


Figure 4.9. Exemple de visualisation pour détecter des valeurs aberrantes

disponibilité et un accès facile, rapide et sécurisé à un grand volume de données variées telles que les données d'auscultation, les paramètres de sol et les paramètres de pilotage du tunnelier. L'intégrité ou la fiabilité des données est garantie par le choix d'une technologie éprouvée, PostgreSQL, qui est un système de base de données relationnelle open source. Son architecture relationnelle permet de mettre en place des relations entre les différentes entités (tables), permettant ainsi une organisation optimale des données sans redondance. Le stockage est ainsi minimisé et l'interrogation de la base de données par des requêtes SQL est rapide et économe en mémoire vive. De plus, le stockage de la base de données peut être réalisé sur des services de stockage en ligne (Cloud) ce qui permet de partager et de sécuriser les données.

Architecture de la base de données

La première étape dans la mise en place d'une base de données est la conception de son architecture. Cette étape est itérative et chronophage car elle nécessite un consensus entre les experts du domaine et les informaticiens. De plus, il est primordial de concevoir une architecture de base de données qui s'adapte non seulement aux besoins actuels, mais qui soit également capable de répondre aux exigences futures. Autrement dit, dans le cas étudié ici, la base de données doit être en mesure d'intégrer des données provenant de projets ultérieurs de tunnels creusés au tunnelier.

La conception d'une base de données pour un projet de tunnelier est une tâche complexe qui nécessite une réflexion approfondie en raison de l'aspect spatial et temporel lié à l'avancement du creusement du tunnel. Prenons pour exemple des paramètres de

pilotage du tunnelier. Ceux-ci sont mesurés au niveau du front ou à l'arrière de la jupe, pendant l'excavation d'un anneau ou pendant la pose du revêtement. La pression au front s'applique au niveau de la RDC au moment du creusement, l'injection de mortier est réalisée à l'arrière de la jupe, soit environ 14 m devant. Il y a donc deux positions dans l'espace et deux instants différents à prendre en compte. Pour résoudre cette difficulté, il est nécessaire de créer une table intermédiaire, que nous avons appelé ***Anneau***, qui permet de connecter les différents éléments en termes d'espace et de temps. Ainsi, la finalité est de faire en sorte qu'une requête sur la base de données soit capable de fournir des informations précises sur la position, telles que les coordonnées, la stratigraphie et les paramètres mécaniques des sols, les capteurs installés ainsi que leurs mesures, et enfin les paramètres de pilotage du tunnelier. On pourra donc extraire simplement, par exemple, la pression au front lorsque la roue de coupe est à position donnée, ou encore la pression d'injection pour une position donnée de la roue de coupe (sans qu'il soit nécessaire de penser à appliquer un différentiel de 14 m entre position de la roue de coupe et fin de la jupe), etc. Il est essentiel que la base de données soit capable de fournir ces informations de manière claire et concise afin de faciliter l'analyse et l'interprétation des données collectées.

La structure de la base de données choisie dans notre étude est similaire à celle présentée par Marinos et al. (2013). La base est constituée de trois catégories principales et d'une catégorie supplémentaire : (a) les mesures d'auscultations en surface, (b) les paramètres de creusement du tunnelier, (c) les données géologiques et géotechniques et (d) les tables d'assistance qui contiennent les abréviations, symboles, unités, etc. La Figure 4.10 présente une vue synthétique de la structure de la base de données conçue dans cette étude. Ce diagramme entité-relation (DER) est un outil largement utilisé pour la conception de bases de données relationnelles. Il s'agit d'un modèle de représentation graphique dans lequel les entités (tables), les relations et les attributs (colonnes, propriétés de l'entité) sont représentés sous forme de symboles et de lignes. Le DER permet de visualiser clairement et de manière concise les relations entre les entités. Les détails concernant les mises en relation sont décrites dans à la fin de ce §.

La base de données est constituée à ce stade de 12 entités, les principales comportent notamment ***Secteur***, ***Anneau*** et ***Parametre_Mecanique_Formation*** ainsi que les entités d'assistances ***Definition_Parametre_Mecanique_Formation*** (nom, unité physique, symbole et catégorie de chacun des paramètres mécaniques de sol) et ***Formation*** (codification et couleur pour les représentations). Le projet, définie dans la table ***Projet*** (informations telles que le responsable, la localisation, etc.), est divisé en plusieurs secteurs qui définissent la géologie et les paramètres géotechniques du terrain (***Parametre_Mecanique_Formation***). Cette table est alors « parent » à la table ***Stratigraphie*** afin d'obtenir les informations géotechniques de chacune des couches à chaque position sans répétitions des valeurs. La table ***Anneau***, placée au centre de l'architecture, contient la position du tunnelier tous les mètres dans les trois directions de l'espace. Elle est parent aux trois catégories principales : les tables ***Capteur***, ***Parametre_Tunnelier***, ***Stratigraphie***.

Il est ainsi possible d'obtenir la position des mesures de tassements, des paramètres de pilotage du tunnelier et de la stratigraphie traversée dans une requête unique. ***Parametre_Tunnelier*** contient environ 40 paramètres différents et ***Capteur*** est parent à la table ***Mesure_Capteur*** contenant toutes les mesures de tassements récupérées des deux lignes.

Construction de la base de données

Dans le modèle relationnel adopté pour notre étude, chaque table est dotée d'une clé primaire (Primary Key, PK) qui permet d'identifier de manière unique chaque enregistrement dans la table. Les clés primaires sont souvent représentées par une colonne spécifique, souvent numérique pour des raisons de performance lors des jointures, telle que l'*id*, qui ne peut pas avoir de valeurs en double. Par exemple, la clé primaire de la table ***Anneau*** est ***id_anneau***. Attention, le nom de cette colonne fait référence à l'aspect unique de chaque ligne de la table mais ne renvoie pas à un numéro d'anneau comme le nom peut le laisser à penser. Les ***id_anneau*** sont en réalité un indicateur de la position linéaire du tracé, tronçon par tronçon. Contrairement à PK, nous nous sommes fixé la contrainte qu' ***id_anneau*** est obligatoirement croissant dans le sens du creusement des tunnels, ce qui facilite l'exploitation de cette table. Il convient de mentionner à présent que l'injection des données dans cette table a été faite selon l'ordre suivant des tronçons : TR1, TR2, TR3 puis L14S2.

Les tables sont liées entre elles par des clés étrangères (Foreign Key, FK), qui sont les colonnes d'une table faisant référence à la clé primaire d'une autre table. Les relations entre les tables sont établies en fonction des clés primaires et étrangères, permettant ainsi de lier des données d'une table à une autre. Cette architecture en relation permet de faciliter les jointures et les extractions couplant les données de multiples tables. De plus, les clés primaires et étrangères assurent l'intégrité des données. Par exemple, la table ***Mesure_Capteur*** ne peut pas contenir, par construction, des mesures de tassement qui ne sont pas associées au capteur correspondant dans la table ***Capteur*** (sinon, la base renvoie une erreur).

La base de données a été conçue pour être directement utilisable par l'utilisateur. Pour cela, les relations entre les tables doivent être explicites. Ainsi, l'injection de données brutes dans cette base est impossible. Il est nécessaire de créer au préalable des colonnes de liaison entre les tables en question, qui serviront de clés primaires et clés étrangères.

La construction de la base de données doit être réalisée uniquement à l'aide de scripts SQL (Annexe B) et non pas par l'injection de fichiers de type CSV qui ne permettent pas de définir le type des attributs et ne garantissent pas la reproductibilité de la base de données en cas de problème. D'autres recommandations d'ordre pratique sont données dans la dernière partie de cette thèse (§ 7.2.2).

A l'issu de ce travail, la base de données est définie et prête à être interrogée pour l'analyse des données collectées et l'application d'algorithmes d'apprentissage automatique

visant à prévoir les déplacements à l'avant du front du tunnelier. Cependant, avant de procéder à toute analyse de prévision, il est essentiel de quantifier la qualité des mesures obtenues pour chaque capteur. Cette première étape permettra d'assurer la validité et la fiabilité des données utilisées pour l'analyse et de garantir que les résultats obtenus seront suffisamment précis et significatifs au regard des grandeurs en jeu. En d'autres termes, ce travail permet de prévenir les conséquences auxquelles la fameuse formule « garbage in, garbage out » nous enjoint à veiller.

Conclusion

Ce chapitre avait pour objectif de décrire les lignes L14S2 et L15SO du GPE, en mettant notamment en avant les profils géologiques rencontrés, ainsi que de présenter les différentes étapes du traitement spécifique des données collectées pour ces deux lignes. Ces travaux ont été divisés en trois parties : l'extraction des données, le nettoyage et leur stockage dans une base de données.

Dans un premier temps, l'extraction consiste à acquérir les données de sol, de mesures de tassement et de pilotage du tunnelier depuis différents fournisseurs de données. Dans le but d'atteindre cet objectif, des techniques de web scraping et d'OCR ont été mises en œuvre.

Dans un second temps, un nettoyage a été effectué sur l'ensemble des données : organisation, gestion des données manquantes, suppression des données dupliquées et détection des données aberrantes sont les grands titres de cette partie. Néanmoins, il convient de mentionner que des travaux de nettoyage supplémentaire effectués sur les mesures de tassement seront présentés dans le chapitre suivant (§ 5.2.1).

Enfin, nous avons discuté de l'intérêt d'une base de données avant d'introduire son architecture et sa construction. A ce stade, nous avons enfin une base de données opérationnelle avec au centre de son architecture une table ***Anneau*** dont la clé primaire ***id_anneau*** représente la position à chaque mètre du tracé du tunnel. Cette table est reliée à la nature des sols (table ***Stratigraphie***), aux paramètres de pilotage du tunnelier (table ***Paramètre_Tunnelier***) et aux capteurs (table ***Capteur***) à travers l'attribut ***id_anneau*** qui est une clé étrangère dans chacune de ces tables.

Ce travail laborieux et de longue haleine, mais néanmoins riche d'enseignements, a été nécessaire afin de transformer les données brutes en informations exploitables dans le but d'en tirer une valeur ajoutée à travers la modélisation avec l'apprentissage automatique.

Le chapitre suivant sera consacré à l'analyse exploratoire des données collectées ainsi qu'à l'ingénierie des caractéristiques, en vue de la préparation pour l'application des algorithmes d'apprentissage automatique.

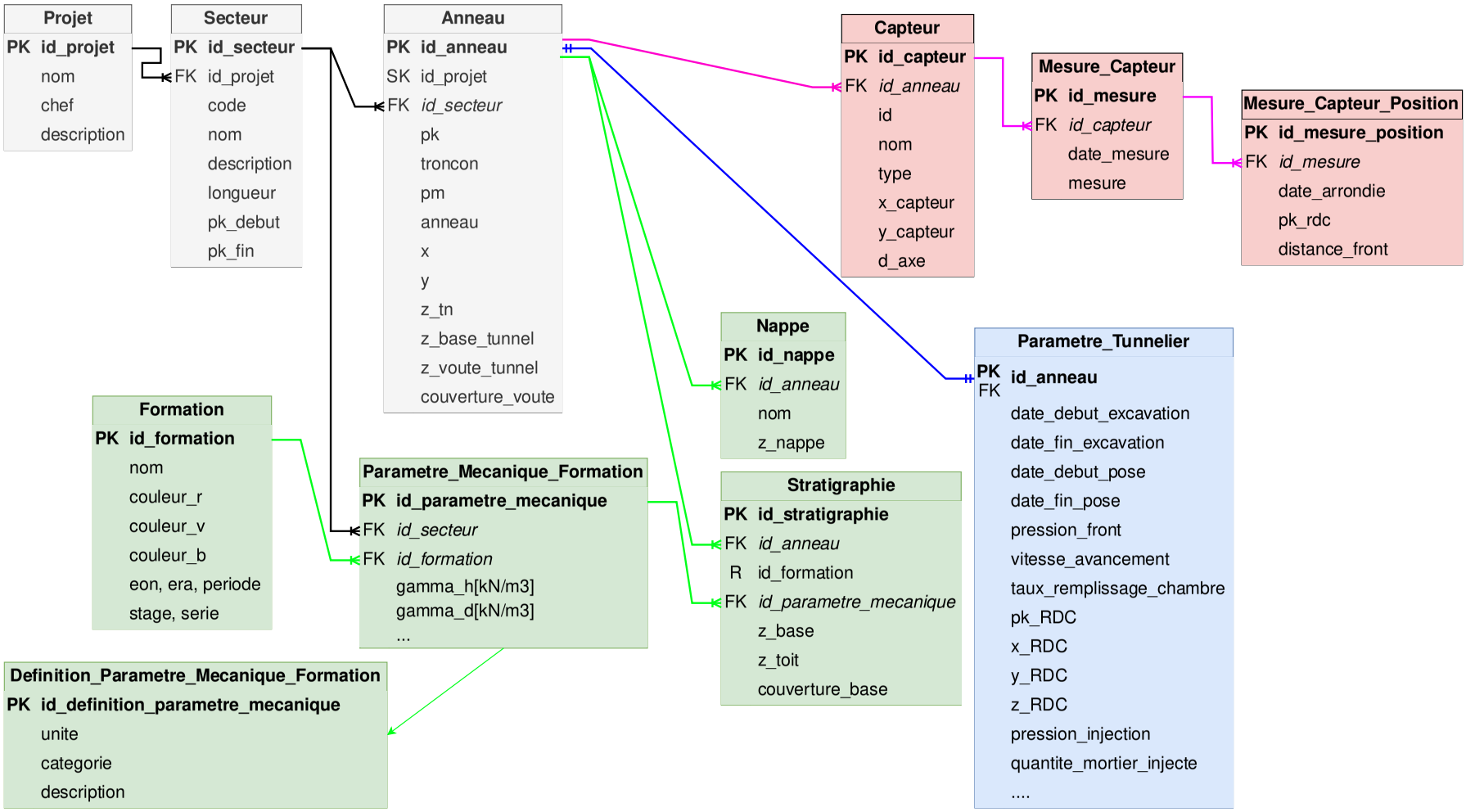


Figure 4.10. Diagramme d'entité-relation de la base de données

Analyse exploratoire et ingénierie des caractéristiques

5

Introduction

Le chapitre précédent a présenté la transformation de la donnée brute en information exploitable. A ce stade, il est nécessaire de faire parler les données en menant des analyses exploratoires. Ensuite, dans le but d'appliquer des algorithmes d'apprentissage automatique pour la prévision des tassements, il est indispensable de sélectionner et d'extraire les caractéristiques nécessaires.

Dans la pratique, ces étapes ne se déroulent pas de manière linéaire, mais plutôt de manière simultanée et interconnectée. Le but de ce chapitre est donc de présenter ces travaux dans l'ordre suivant : tout d'abord, nous réexaminerons la problématique de cette étude en donnant une vue d'ensemble du problème ; puis nous enchaînerons sur l'extraction des variables nécessaires pour l'apprentissage automatique ; enfin, nous présenterons des analyses statistiques sur les données.

5.1 Vue d'ensemble du problème

Avant de procéder aux analyses statistiques, il est crucial de revenir à l'objectif final de cette étude : quelles sont les variables que l'on veut prévoir et quels sont les paramètres pertinents à considérer pour atteindre cet objectif ? Cette section répondra de manière rigoureuse à ces deux questions afin d'établir des fondations solides pour la suite des travaux.

5.1.1 Choix des variables cibles

Cette thèse se consacre à la prévision des tassements en surface induits par le creusement au tunnelier. Toutefois, ce problème est un phénomène tridimensionnel complexe, caractérisé par plusieurs paramètres tels que le tassement maximal le long de l'axe du tunnel (s_{max}, § 1.2.2), le tassement maximal à une distance donnée de l'axe du tunnel (s^*, § 1.2.2) ou encore le tassement à une certaine distance du front du tunnel (s_{long}, § 1.2.2). Dans l'état de l'art, les études ont cherché à prévoir majoritairement s_{max} mais aussi s_{long} et s^* (Table 3.2, § 3.4.1).

Les données obtenues dans le cadre du GPE contiennent des mesures de tassements effectuées toutes les 30 minutes environ (§ 4.2.1). Ces informations permettent d'observer la progression du tassement en un point fixe de l'espace en fonction de l'avancement du tunnelier par rapport à ce point. Différentes manières d'aborder le problème sont possibles. Une piste de travail aurait été de tenter d'utiliser directement ces mesures, sans travail

préalable de paramétrisation de la forme des déformations de surface. Ce n'est pas le choix qui a été fait pour ce travail de thèse qui se concentre sur la prévision des paramètres de la cuvette, et plus particulièrement de s_{max} et s^*, issus de calages préalables. C'est un choix fort, motivé par des considérations pratiques d'ingénieur (compréhension, découpage, simplification du problème à traiter, recombinaison), qui donne une direction claire à l'étude et qui permet le contrôle de toutes les étapes nécessaires à la prévision.

Pour obtenir les valeurs de s^*, il faut caler les équations de progression du tassement sur les mesures observées par chacun des capteurs. Ensuite, on cale sur ces s^* l'équation du tassement transversal, ce qui permet d'obtenir les valeurs de s_{max} tout au long du tracé (Figure 5.1). Les paramètres calés sont ensuite injectés dans deux nouvelles tables dans la base de données :

- la première a comme clé primaire ***id_anneau*** qui sert également de clé secondaire pour connecter cette nouvelle table à la table ***Anneau***. Cela permet d'obtenir le tassement transversal en fonction de la position ;
- la deuxième a comme clé primaire ***id_capteur*** qui sert également de clé secondaire pour connecter cette nouvelle table à la table ***Capteur***. Cela permet d'obtenir la courbe de progression du tassement pour chacun des capteurs et ainsi pour chaque position par rapport à l'axe du tunnel.

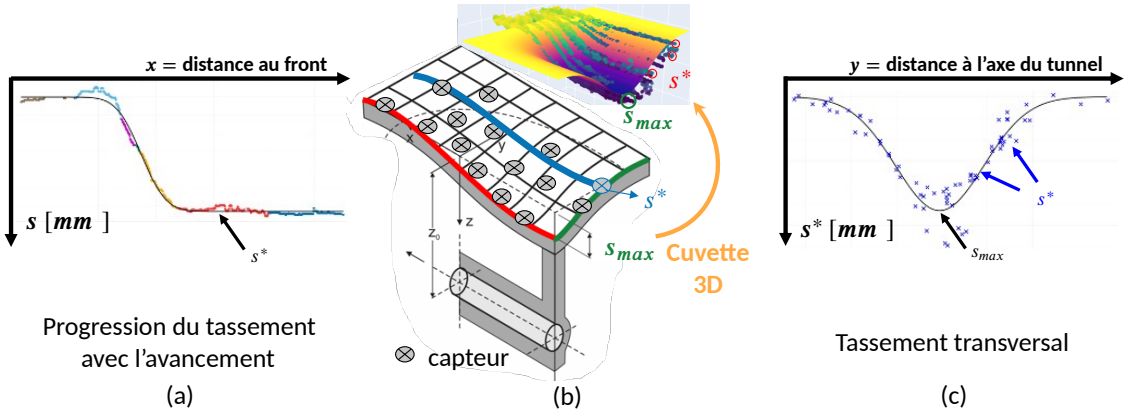


Figure 5.1. Calage de l'équation de progression du tassement sur les mesures brutes afin d'obtenir s^* en un point fixe (a) puis utilisation des valeurs de s^* obtenues en tout point (b) pour caler l'équation transversale du tassement et obtenir ainsi la valeur de s_{max} (c)

5.1.2 Sélection des caractéristiques

Les algorithmes d'apprentissage automatique prévoient la variable cible à partir de caractéristiques. La sélection des caractéristiques les plus pertinentes est donc cruciale pour obtenir des modèles performants. Pour ce faire, on s'inspire de l'état de l'art (Figure 3.6) ainsi que de l'expérience métier. A ce stade, on choisit les paramètres suivants sur lesquels on effectuera des analyses statistiques : paramètres de géométrie (couverture au-dessus de la voûte du tunnel C [m]), paramètres de pilotage du tunnelier (vitesse d'avancement

$V_{tunnelier}$ [mm/min], moment de la roue de coupe M_{RDC} [kN.m], pression au front P_{front} [bar], pression $P_{mortier}$ [bar] et quantité de mortier injecté $V_{mortier}$ [m^3], poussée totale du tunnelier P_{totale} [kN]) et paramètres géologiques et géotechniques (poids volumique γ [kN/m^3], module pressiométrique de Ménard E_M [MPa], coefficient rhéologique α, cohésion c [kPa], angle de frottement φ [°] et coefficient de pression des terres au repos K_0). À cette liste de caractéristiques éventuelles, on ajoute la distance des capteurs à l'axe du tunnel d_{axe}. Le calcul de ce paramètre, ainsi que de la distance du capteur au front du tunnel d_{front}, est présenté dans la section suivante. Il est à signaler qu'en première approche, la présence des nappes phréatiques n'est pas prise en compte dans nos études. A l'issue de ces analyses, les caractéristiques à prendre en compte dans les algorithmes d'apprentissage automatique sont sélectionnées parmi ces paramètres.

Réduction des dimensions des paramètres de sol

En premier lieu, il est important d'aborder la question de la dimension des paramètres de sol. Comme expliqué précédemment dans le § 3.3.2, la définition correcte d'une stratigraphie nécessite une multitude de paramètres géologiques. Afin de réduire la dimensionnalité des paramètres sélectionnés précédemment, on choisit d'utiliser la méthode de combinaison des paramètres de sol proposée par Chen et al. (2019a) (Équation 3.1). Nous avons toutefois choisi de considérer, pour notre cas d'étude, les couches de sol jusqu'à la base du tunnel et non pas jusqu'à la voûte comme l'auteur le propose. Les modifications introduites sont mises en évidence dans la Figure 5.2.

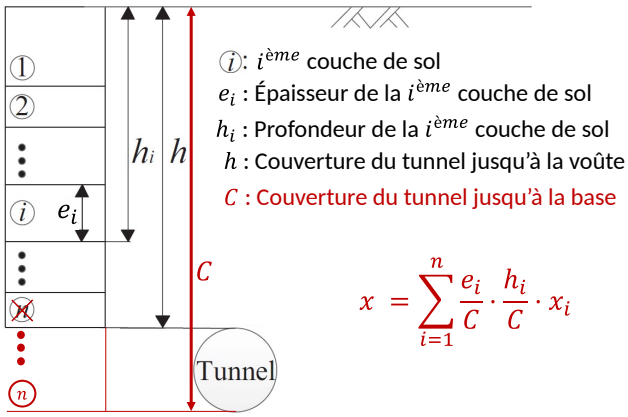


Figure 5.2. Schématisation de la méthode de combinaison des paramètres de sols. Les modifications par rapport à la méthode initiale (Chen et al., 2019a) sont marquées en rouge.

La Figure 5.3 permet d'illustrer la valeur des paramètres transformés le long des tracés. Les valeurs de s_{max} calées dans la section suivante sont également présentées. On peut constater en particulier que les valeurs extrêmes de s_{max} sont bien observées dans les zones avec les plus faibles couvertures, ce qui est cohérent avec nos connaissances métiers. Ensuite, on remarque que les zones « homogènes » (stratigraphie et couverture similaires)

ont des paramètres constants. Les variations de paramètres sont dues à des changements de stratigraphie, de couverture ou bien de secteur et donc de paramètres mécaniques de sol. En effet, le changement de secteur peut induire des décrochements parfois brutaux, mais cela est normal au regard des choix qui ont été faits. Une piste d'amélioration de la technique dans notre cas serait de pouvoir lisser ces discontinuités. Il convient de noter que cette méthode est une transformation qui ne conserve pas l'intervalle des données d'entrée. A ce titre, c'est une transformation destructrice du sens physique des paramètres, et non pas une homogénéisation au sens strict. Ce n'est pas pour autant un problème pour une utilisation dans des algorithmes d'apprentissage, car ce sont les variations d'ensemble qui importent.

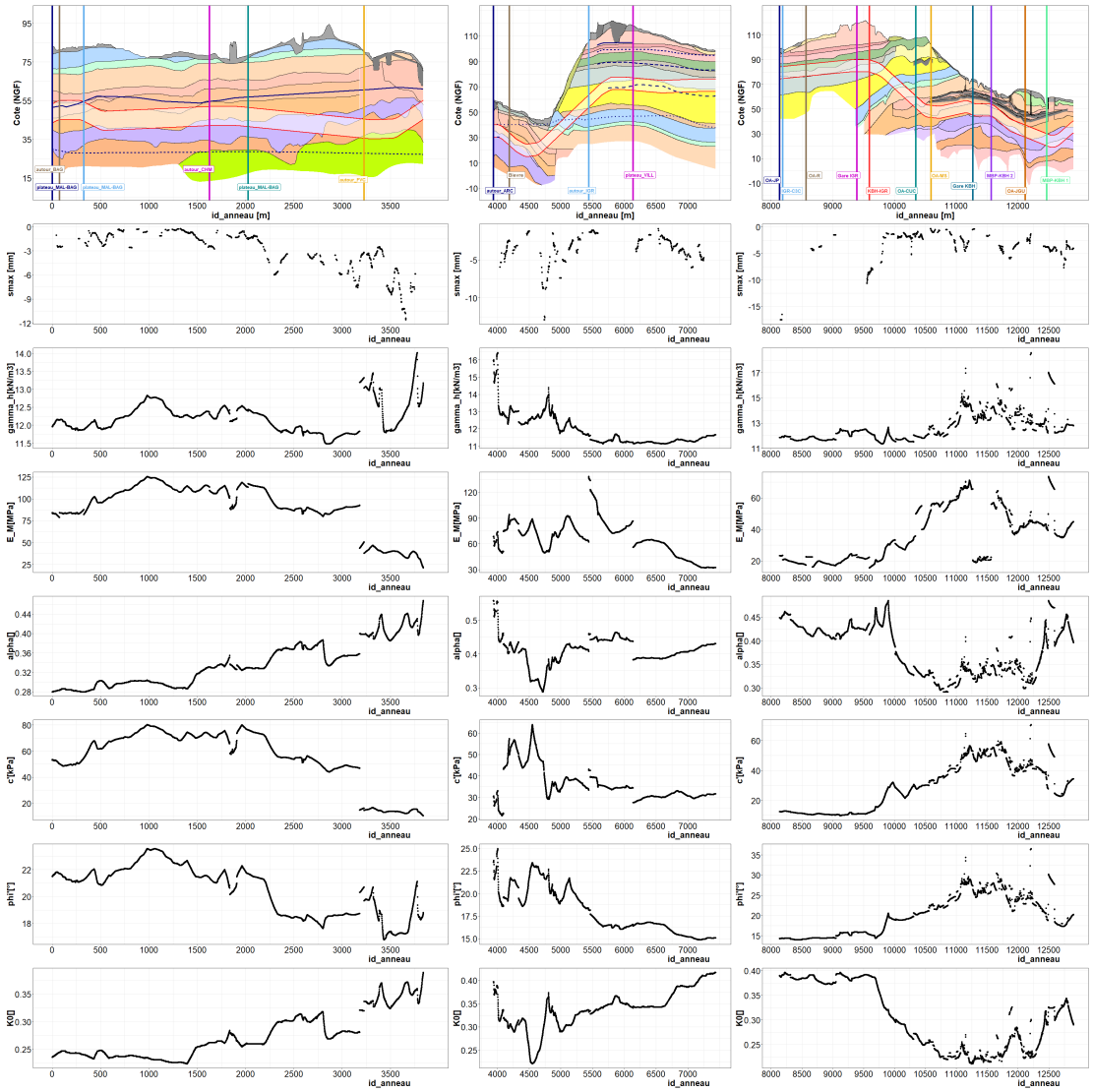


Figure 5.3. Distribution des s_{max} et des paramètres mécaniques combinés du sol en fonction du profil en long. Les lignes verticales sur le profil en long indiquent le changement de secteur et par conséquent le changement des paramètres mécaniques des couches. Ordre des figures de gauche à droite : TR1, TR2 et L14S2.
Pour la légende des sols, se référer à la Figure 4.1

Calcul de la distance à l'axe

La distance des capteurs par rapport à l'axe du tunnel (d_{axe}) est essentielle pour caler par la suite la cuvette transversale de tassement. Cependant, cette information n'est pas fournie par les propriétés des capteurs et doit donc être calculée. Pour cela, on utilise les coordonnées des capteurs ainsi que celles du tracé du tunnel. La convention de signe adoptée pour d_{axe} est telle que les valeurs sont positives dans la région située à l'est de la ligne L14S2 et au sud de la ligne L15SO, comme indiqué dans la Figure 5.4. Il convient de noter que le signe de d_{axe} n'est pas défini par rapport à la direction de l'axe y, telle que décrite dans le § 1.2.2 car le creusement des tronçons ne suit pas un sens unique.

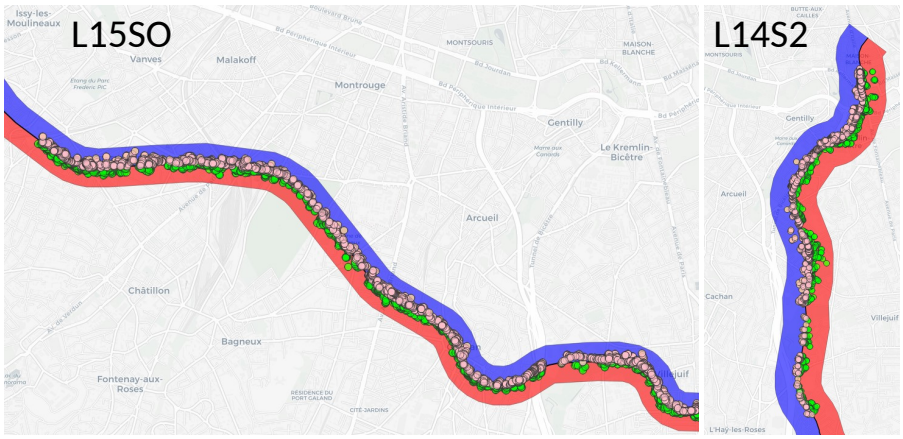


Figure 5.4. Convention de signe de la distance des capteurs à l'axe au tunnel (d_{axe} positive dans la zone en rouge sur L14S2 et L15SO)

Calcul de la distance au front

Lors d'un fonctionnement normal, les théodolites effectuent des visées sur les capteurs environ toutes les 30 minutes ou toutes les heures, ce qui permet d'obtenir des mesures de tassement en fonction du temps (Table A, Figure 5.5). Cependant, l'équation décrivant le tassement longitudinal (Équation 1.10) est établie en fonction de l'espace plutôt que du temps. Il est donc nécessaire de calculer la position du front du tunnel à la date des mesures de tassement.

La position du front pk_{rdc} est une information connue à partir des paramètres relatifs au pilotage du tunnelier. En effet, pk_{rdc} est fournie aux dates correspondant au début (DDE) et à la fin (DFE) de l'excavation d'un anneau, ainsi qu'au début (DDP) et à la fin (DFP) de la pose de ce dernier. La durée d'une excavation varie entre 1 heure 30 minutes et 3 heures environ, et la durée de pose est d'environ une demi-heure. Seules les DDE et DFE sont utilisées (Table B dans la Figure 5.5) puisque la position du front varie uniquement en phase d'excavation. Les étapes permettant de retrouver la position du front du tunnel à la date des mesures de tassement sont décrites dans la Figure 5.5.

A l'issue de ce travail, il est possible d'obtenir des visualisations des mesures de

tassement en fonction de la date et de la distance au front du tunnel comme illustré dans la Figure 5.6

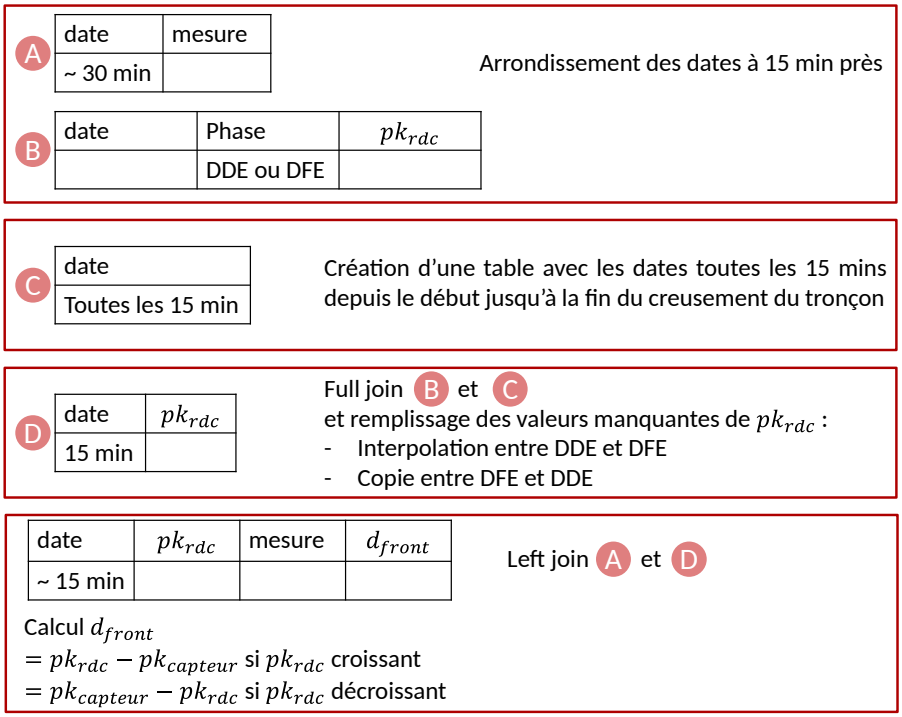


Figure 5.5. Étapes suivies pour le calcul de la distance du capteur au front du tunnel. Ces étapes sont répétées pour chaque tronçon.
Remarque : Les types de jointure sont expliqués dans l'Annexe C.

5.2 Calage des équations de tassement

Pour obtenir les variables cibles s_{max} et s^*, il faut passer par un calage des équations de progression du tassement sur les mesures brutes observées par chacun des capteurs et ensuite caler l'équation du tassement transversal sur les s^* obtenus. Cependant, afin de réduire les pertes et d'obtenir des calages suffisamment fiables et précis, il faut commencer par nettoyer les mesures de tassements au-delà de ce qui a déjà été fait lors du nettoyage « primaire » présenté au § 4.2.2.

5.2.1 Nettoyage des mesures de tassement

Le nettoyage supplémentaire appliqué sur les mesures de tassements consiste à appliquer des filtres sélectifs, à éliminer les mesures aberrantes et à réduire le bruit des mesures.

Filtrage

Une série de sélections par filtrage ont été effectuées afin d'augmenter la qualité des données avant de lancer des études supplémentaires :

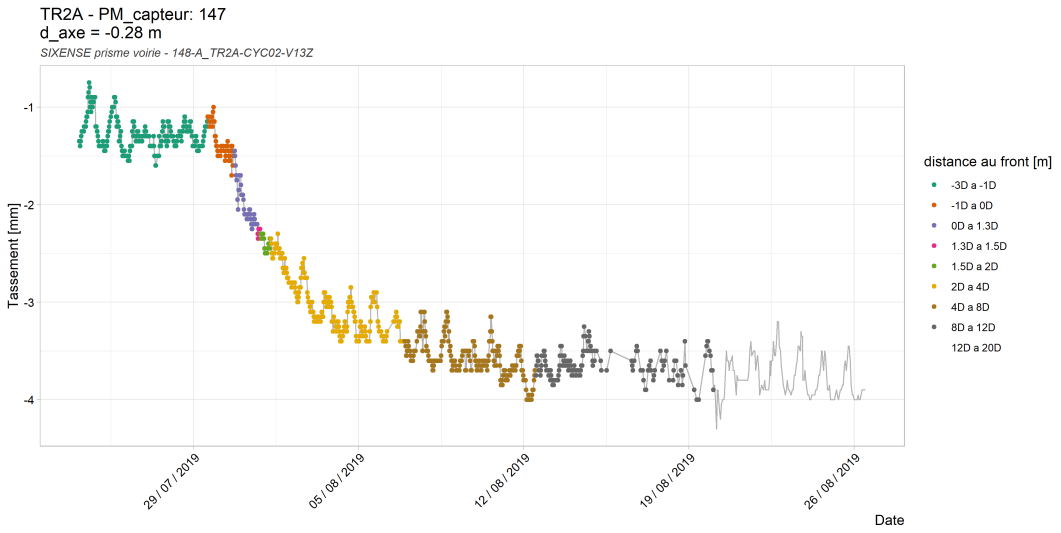


Figure 5.6. Exemple de mesures de tassement en fonction de la date et de la distance au front du tunnel

1. Filtre sur les types de capteurs à retenir.
   Seuls les capteurs placés sur les voiries ou les bâtiments (en bas ou en haut) ainsi que les cibles virtuelles ont été injectés dans la base de données (capteurs de type : Centaure, Cible, Prisme, Prisme Voirie, Prisme Bâti, Prisme Bas et Prisme Haut). A ce stade, on obtient une table ***Capteur*** avec 8 689 capteurs et une table ***Mesure_Capteur*** avec 67 935 713 mesures de tassements associés. La répartition de ces capteurs sur les tracés des lignes L14S2 et L15SO est montrée dans la Figure 5.7.

2. Filtre sur la distance des capteurs au front du tunnel (d_{front}).
   Il convient de noter que des mesures coïncident en termes de dates avec une absence de creusement, par exemple au début et à la fin des tronçons. Ces valeurs sont forcément éliminées.
   On cherche également à éliminer les mesures de déplacement qui ne sont pas induits par le creusement. Nous avons donc gardé uniquement les mesures obtenues 150 m avant le passage du tunnelier et jusqu'à 500 m après le passage du tunnelier. Cela correspond à environ 10 à 15 jours avant le passage du front du tunnel et 1 mois après (§ 5.3.1). Ce filtre simple permet d'épurer une grande quantité de données : le nombre de mesure est divisé par 5 (13 141 328 observations restantes). En termes d'espace mémoire pour le stockage, on passe d'une taille initiale de 4.7 Gb à une taille de 1 Gb, ce qui permet de gagner largement en performance lors des traitements subséquents.
   A ce stade, une nouvelle table est créée dans la base de données sous le nom de ***Mesure_Capteur_Position*** qui est reliée à la table ***Mesure_Capteur***. Cette table contient alors les mesures épurées en fonction de la position du front (pk_{rdc}) et de la distance au front (d_{front}).
   Par la suite (§ 5.3.1), nous verrons pourquoi il s'est finalement avéré suffisant de

travailler avec un intervalle entre 100 m avant le passage du front du tunnel jusqu'à 250 m après le passage du tunnelier.

3. Filtre sur la distance des capteurs à l'axe du tunnel (d_{axe}).
   Les capteurs à une distance $d_{axe} \geq 7 \times D$ sont supprimés, soit 68 capteurs.
4. Filtre pour supprimer les capteurs qui n'ont pas suffisamment de mesures pour permettre de caler l'équation de progression du tassement (Équation 1.10). Pour réussir le calage, après une série de tests, nous avons choisi de retenir le seuil d'au moins 10 mesures entre $d_{front}= -50$ m et $d_{front}= 0$ m et d'au moins 10 mesures entre $d_{front}= 30$ m et $d_{front}= 250$ m.

A ce stade, la table contient 8 706 685 observations issues de 6 799 capteurs.

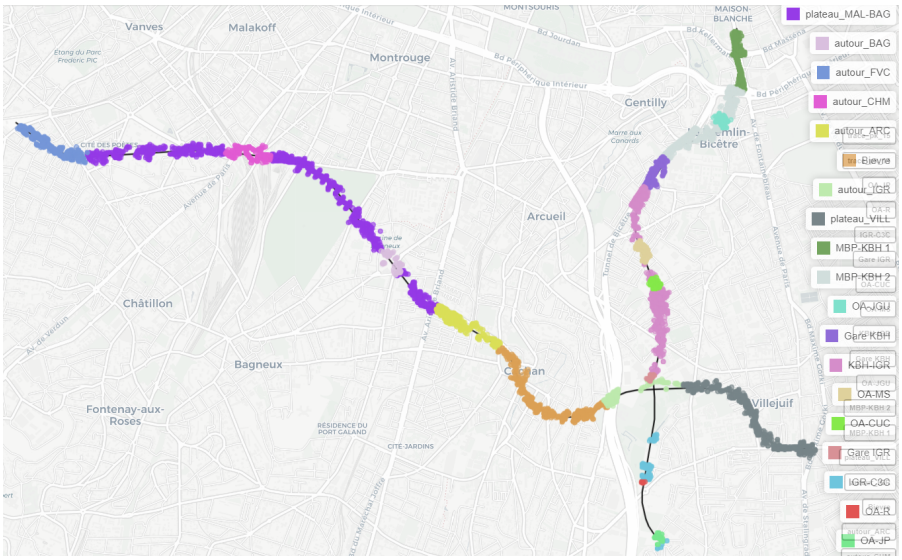
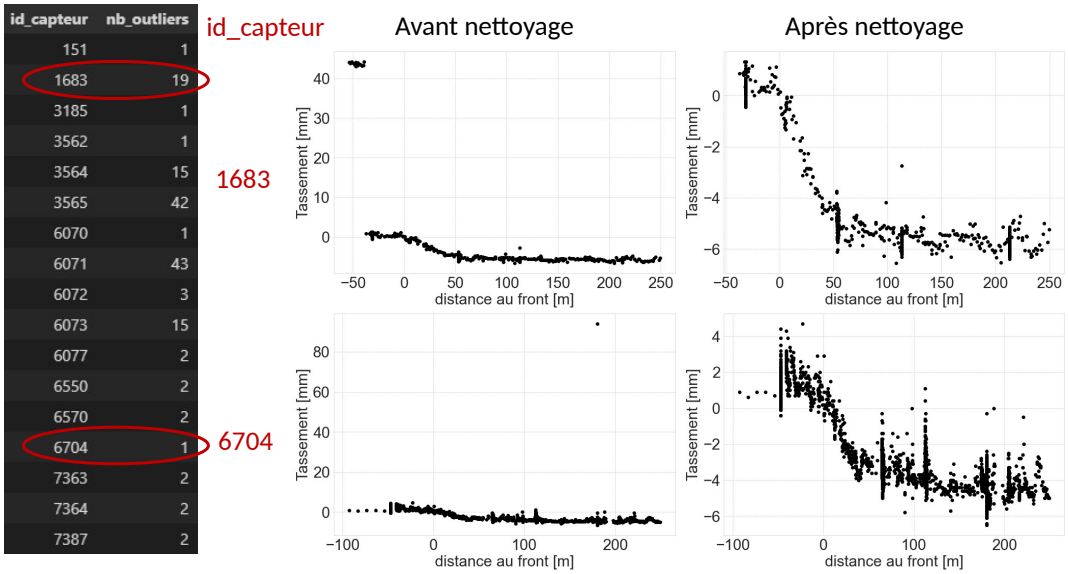


Figure 5.7. Répartition des capteurs sur les lignes L14S2 et L15SO. Les couleurs indiquent les secteurs

Données aberrantes

Deux méthodes sont appliquées pour supprimer les mesures aberrantes. La première consiste à appliquer un filtre simple pour supprimer les valeurs les plus atypiques. Ce filtre consiste à calculer la moyenne des mesures de tassement d'un capteur (s_{moy}) et supprimer toute valeur qui n'appartient pas à l'intervalle $[s_{moy} - 30\ ;\ s_{moy} + 30]$. La valeur de 30 mm est choisie au regard des tassements maximaux observés sur les deux lignes L14S2 et L15SO qui ne dépassent jamais les 20 mm. Le nombre de capteurs affectés par ce filtre est présenté dans la Figure 5.8, qui montre par ailleurs quelques exemples de données avant et après application de ce filtre. Au total, 154 mesures sont supprimées.

La seconde méthode est l'utilisation de l'algorithme des forêts d'isolation IF (§ 2.3.1). Ce dernier est un algorithme d'apprentissage automatique bien adapté à l'apprentissage non-supervisé, en particulier pour les tâches de détection d'anomalies. Il suffit de prendre la distance au front et les mesures en tant que paramètres d'entrée et l'algorithme retourne



id_capteur	nb_outliers
151	1
1683	19
3185	1
3562	1
3564	15
3565	42
6070	1
6071	43
6072	3
6073	15
6077	2
6550	2
6570	2
6704	1
7363	2
7364	2
7387	2

Figure 5.8. Nombre de mesures de tassement éliminées lors du filtre ± 30 mm, et exemples de données de capteurs filtrés

en sortie une note pour chaque mesure indiquant ainsi si cette dernière est typique ou aberrante. Ce filtre a permis d'éliminer 175 856 mesures aberrantes. Quelques exemples de résultats issus de l'application des forêts d'isolation sont présentés dans la Figure 5.9. Il convient de noter que l'application du premier filtre simple est nécessaire avant l'utilisation des forêts d'isolation car ce dernier se base sur la détection de groupement de données. Par conséquent, des mesures comme celles observées par le capteur 1683 (Figure 5.8) n'auraient pas été considérées par l'IF comme des mesures aberrantes vu qu'elles forment un groupement.

```python
# {python}
cpt_list = all_data.id_capteur.drop_duplicates().values
for cpt in cpt_list:
    df = all_data[all_data.id_capteur == cpt]

    # selection des caracteristiques
    anomaly_inputs = ['d_front', 'mesure']

    # contamination: valeur entre 0 et 0.5, a augmenter pour avoir un modele plus strict.
    # random_state: pour avoir des resultats reproductibles.
    model_IF = IsolationForest(contamination = float(0.02), random_state = 42)

    # Entrainement de l'algorithme
    model_IF.fit(df[anomaly_inputs])

    # obtention des predictions
        # anomaly_scores : score de chaque observation dans l'ensemble des donnees.
```

```
18            # Plus le score est bas, plus l'observation est anormalle.
    Les valeurs negatives indiquent que l'observation est aberrante.
19            # anomaly : valeur 1 ou -1 pour indiquer si l'observation est
    typique ou aberrante.
20     df['anomaly_scores'] = model_IF.decision_function(df[anomaly_inputs
    ])
21     df['anomaly'] = model_IF.predict(df[anomaly_inputs])
```

Script 5.1 Entraînement de l'algorithme forêt d'isolation (IF)

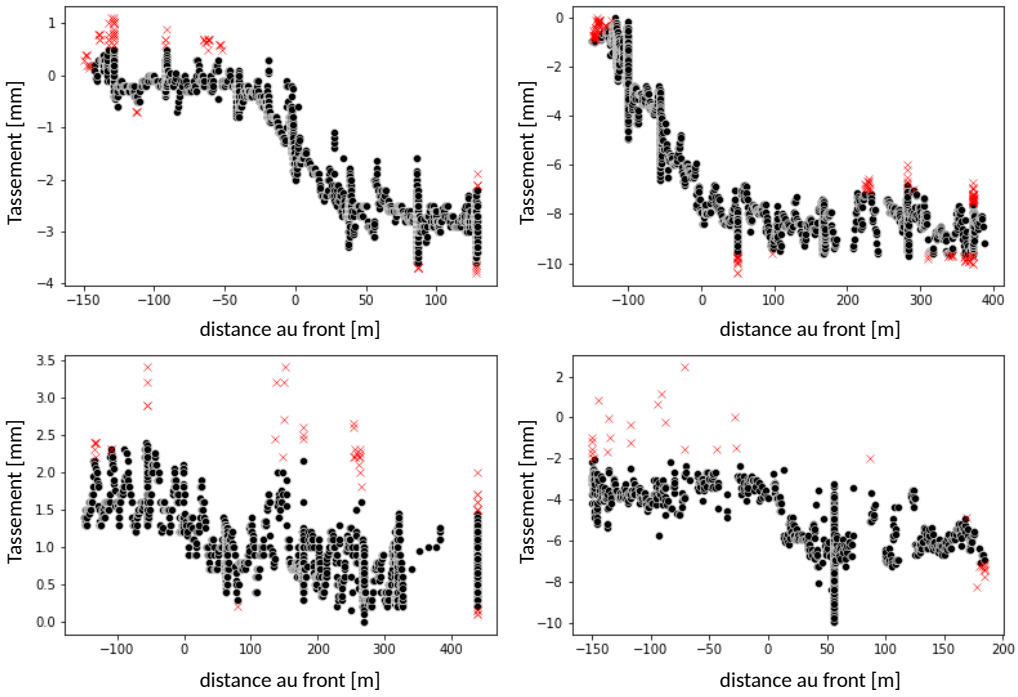


Figure 5.9. Exemples de résultats issus de l'algorithme des forêts d'isolation (IF). Les mesures identifiées comme aberrantes sont signalées en rouge

Données bruitées

Des techniques de lissage de courbes telles que les moyennes et médianes mobiles ont été testées pour éliminer le bruit des mesures de tassement obtenues par un capteur (Figure 5.10). L'objectif ultime de ce nettoyage est de parvenir à caler les paramètres des équations de tassement (§ 5.2.2). Nous avons donc comparé les résultats de calage de s^* sur des mesures brutes ainsi que sur des mesures nettoyées à l'aide de la moyenne mobile sur les distances (Script 5.2). Les résultats présentés dans la Figure 5.11 montrent clairement que le calage par la méthode des moindres carrés réussit aussi bien sur des mesures bruitées que sur des courbes lissées. Nous avons donc fait le choix de ne pas conserver ce travail de lissage des courbes pour la suite.

```
1 # {python}
2 df_moyenne = pd.DataFrame(columns= ["id_capteur", "d_front", "mesure"])
```

```
for id_cpt in liste_capteurs:
    df = all_data[all_data.id_capteur == id_cpt]
    for i in range(-100, 250, 2):
        moyenne = df[(df.d_front > i) &
                     (df.d_front <= i+2)].mesure.mean()
        if not np.isnan(moyenne):
            df_moyenne = df_moyenne.append(
                pd.Series({"id_capteur":id_cpt,
                           "d_front": i, "mesure": moyenne}),
                ignore_index = True)
```

Script 5.2 Moyenne mobile sur une distance de 2 m

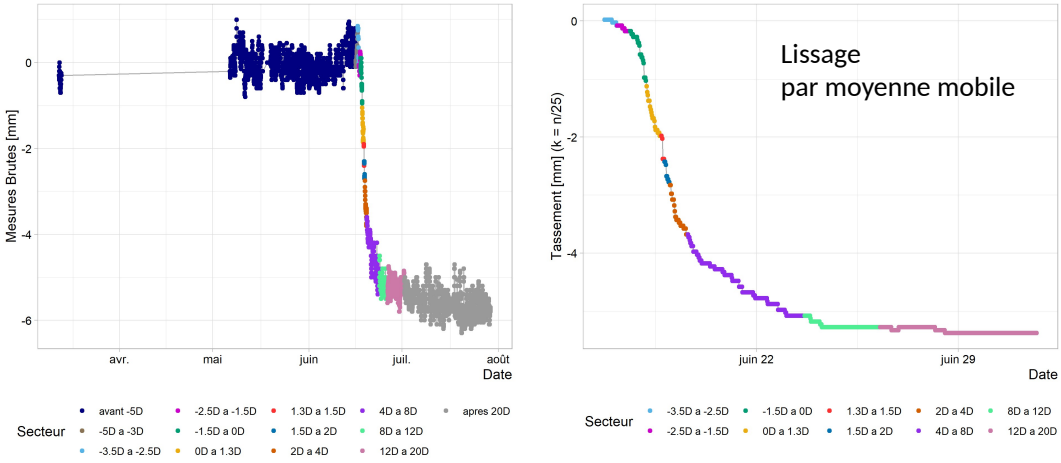


Figure 5.10. Exemples de résultats du lissage des mesures par moyenne mobile, avec largeur de la fenêtre $k = n/25$ où n est le nombre de mesures

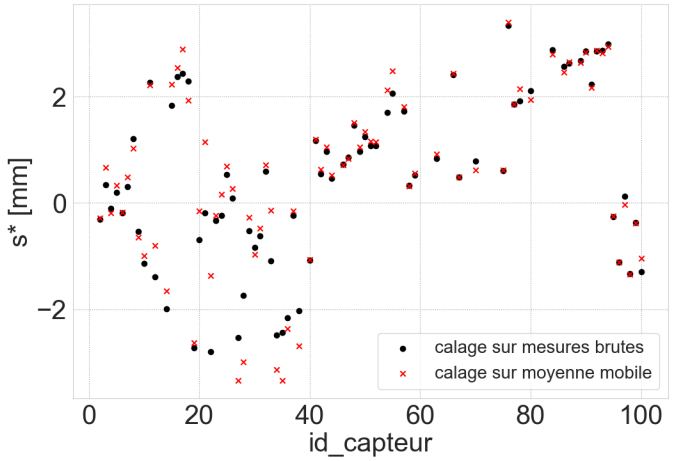


Figure 5.11. Différence du calage de s^* en utilisant les mesures brutes ou les mesures après lissage par moyenne mobil

5.2.2 Équation de progression du tassement

L'étape suivante consiste à caler l'équation de progression du tassement en fonction de l'avancement du creusement (§ 1.2.2, Équation 1.10). Cette équation est paramétrée par les éléments suivants :

- s^* : tassement maximal observé par un capteur (point fixe de l'espace).
- i_x : facteur de forme qui rend compte de la position du point d'inflexion de la dérivée de cette gaussienne cumulée, et donc de la pente maximale de la courbe dans sa zone de transition.
- m_x : coefficient de décalage horizontal de la courbe qui donne une liberté sur la valeur de tassement au passage du front de taille, qui, nous le verrons après (§ 5.3.1), n'est pas toujours égal à la moitié du tassement maximal comme indiqué par Attewell et Woodman (1982).
- s_0 : valeur absolue de décalage vertical de la courbe (offset).
  Les mesures des capteurs sur chantier peuvent être perturbées par de nombreux facteurs externes tels que des problèmes du capteur, des bruits ou vibrations dus au chantier ou à la circulation routière, ou encore la météo (pluie, sécheresse, etc.). Pour éviter de prendre ces facteurs en compte, une remise à 0 du tassement bien avant le passage de la roue de coupe en-dessous du capteur doit être effectuée. Cela est possible en décalant l'équation d'une valeur à caler, notée s_0.

Il convient de rappeler que i_x et m_x sont exprimés en mètre et que leur effet sur la forme de la courbe est présenté dans la Figure 1.11.

Le calage se fait avec la méthode des moindres carrés, à l'aide de la fonction *curve_fit* de la librairie *SciPy* de Python. En plus de l'équation de progression du tassement, de la distance au front d_{front} et des mesures de tassement, cette fonction prend en argument les éléments suivants :

- un vecteur p_0 contenant une estimation initiale des paramètres à caler. Nous avons choisi les valeurs suivantes : $p_0 = [-3, 10, 10, s_{moy}]$ pour s^*, i_x, m_x et s_0, respectivement (les valeurs numériques étant données en mm).
- un vecteur *bounds* indiquant les limites inférieures et supérieures des paramètres à caler. Les limites adoptées (en mm) sont :

$$\begin{aligned} bounds = ([s^*_{inf} = -30, \quad & i_{x_{inf}} = 5, \quad & m_{x_{inf}} = -10, \quad & s_{0_{inf}} = s_{mean} - 30], \\ [s^*_{sup} = 5, \quad & i_{x_{sup}} = 100, \quad & m_{x_{sup}} = 30, \quad & s_{0_{sup}} = s_{mean} + 30]) \end{aligned}$$

  Ces limites, choisies par tâtonnement, sont assez larges pour réduire la perte de capteurs lors du calage tout en optimisant la qualité du calage des courbes. Par exemple, les limites de m_x sont sélectionnées de façon à ne pas prendre en compte les éventuelles variations observées bien avant ou après le passage du tunnelier. Concernant les limites de s^*, nous avons fait le choix de garder une certaine liberté

pour la limite supérieure afin de pouvoir détecter les capteurs indiquant un soulèvement, comme ceux montrés dans la Figure 5.15. Ces soulèvements peuvent être dûs à des facteurs externes tels que la pression d'injection du mortier à l'arrière de la jupe ou bien une pression de front élevée. Par la suite, les capteurs qui mesurent des soulèvement sont supprimés, soit 1061 capteurs (Figure 5.23a).

– une valeur pour *maxfev* qui est le nombre maximal d'appels de la fonction afin de trouver les paramètres optimaux. La valeur par défaut est de 600. S'il s'avère nécessaire d'augmenter grandement ce paramètre, cela indique que le modèle choisi n'est pas le bon. Néanmoins, de façon à obtenir systématiquement des résultats de calage, et donc de ne pas risquer de perdre des mesures par manque d'itérations, et ayant par ailleurs des milliers de courbes à caler automatiquement, nous nous sommes permis de monter cette valeur à 10 000.

```python
# {python}
import pandas as pd
from scipy.optimize import curve_fit
from scipy.stats import norm
# definition de l'equation de progression du tassement
def S_temporel(d_front, Se, ix, mx, s0):
    return Se*norm.cdf(d_front, loc = mx, scale = ix) + s0
                                # loc = moyenne et scale = ecart-type
# definition de la fonction de calage des parametres
def calage_Stemporel(df, id_capteur):
    df = df[df["id_capteur"] == id_capteur].sort_values("d_front")
    X = df["d_front"].values
    Y = df["mesure"].values
    s_moy = Y.mean()
    (Se, ix, m, s0) = curve_fit(S_temporel, X, Y,
                                p0 = [-3, 10, 10, s_moy],
                                bounds = ([-30, 5, -10, s_moy-30],
                                          [5, 100, 30, s_moy+30]),
                                maxfev = 10000)
```

Script 5.3 Calage de l'équation de progression du tassement (Se = s^*, norm.cdf = $G_{[\mu,\sigma]}(\alpha)$, d_{front}, i_x, m_x, s_0)

Par la suite, quelques vérifications sont effectuées telles que l'observation des capteurs avec des $s^* < -15$ mm (généralement, s^* ne dépasse pas des valeurs de l'ordre de 12 mm (en valeur absolue)) ou encore des $|s_0| > 15$ mm (Figure 5.23a). Les paramètres de ces capteurs sont soit corrigés manuellement en testant différentes combinaisons de paramètres, soit supprimés.

L'étape suivante consiste à trouver une méthode pour évaluer la qualité des paramètres obtenus et ainsi trancher sur la « bonne » ou « mauvaise » qualité des capteurs. Pour cela, nous avons calculé l'erreur entre les valeurs mesurées et les valeurs calées en utilisant le cœfficient de détermination R^2 (Équation 2.2). Il convient de noter que le score en question ne permet pas de fournir une réponse définitive : la distinction entre

un capteur de qualité supérieure et un capteur de qualité inférieure demeure subjective et peut être soumise à des critiques. Nous avons fait le choix de garder les capteurs avec un $R^2 \geq 0.02$ afin de garder le plus grand nombre de données possible. Ce choix sera discuté dans la suite de cette thèse compte tenu des résultats obtenus par les algorithmes d'apprentissage automatique (§ 7.2.2). Quelques exemples de calage sont montrés dans les Figures 5.12, 5.13, 5.14 et 5.15.

5.2.3 Équation du tassement transversal

Le calage de l'équation de progression du tassement sur les mesures des capteurs a permis d'obtenir le tassement maximal mesuré de part et d'autre de l'axe du tunnel (s^*). Par suite, il est possible de caler l'équation du tassement transversal en fonction de la distance à l'axe (§ 1.2.2, Équation 1.1). Cette équation est paramétrée par les éléments suivants :

- s_{max} : tassement maximal observé à l'axe du tunnel après stabilisation.
- i_y : distance entre l'axe et les points d'inflexion de la courbe gaussienne.
- m_y : coefficient de décalage transversal qui indique la distance à l'axe à laquelle on observe s_{max}. En théorie, m_y est nul. Toutefois, en pratique, il est conseillé d'ajouter ce degré de liberté pour mieux caler la courbe sur les mesures.

Il convient de rappeler que i_y et m_y sont exprimés en mètre. Comme précédemment, le calage se fait avec la méthode des moindres carrés, à l'aide de la même fonction. Les valeurs des arguments choisis sont les suivantes :

- $p_0 = [\,-2,\ 15,\ 0]$ pour s_{max}, i_y et m_y, respectivement (les valeurs numériques étant données en mm).
- Les limites adoptées (en mm) sont :

$$bounds = ([s_{max_{inf}} = -20, \quad i_{y_{inf}} = 5, \quad m_{y_{inf}} = -5],$$
$$[s_{max_{sup}} = 0, \quad i_{y_{sup}} = 35, \quad m_{y_{sup}} = 5])$$

- *maxfev*$= 5\,000$

La première question qui se pose est celle de la tolérance acceptable permettant de considérer que les s^* font partie de la même section, puisque les capteurs correspondant à une section donnée ne sont pas parfaitement alignés sur une même orthogonale au tunnel. La deuxième question est de savoir à partir de combien de points il est possible de caler une gaussienne sur les mesures. Nous avons fait le choix de prendre une tolérance de 20 m longitudinalement et d'avoir au moins 5 mesures par section. Deux conditions supplémentaires ont été imposées : celle d'avoir au moins une mesure à $|d_{axe}| \leq 5$ m et celle d'avoir au moins une mesure à $|d_{axe}| \geq 15$ m. Ces tolérances et conditions ont été choisies par essais et erreurs. Quelques exemples de cuvettes transversales calées sont montrés dans la Figure 5.16.

```python
# {python}
import pandas as pd
from scipy.optimize import curve_fit
import numpy as np
# definition de l'equation transversale du tassement
def S_trans(d_axe, Smax, iy, my):
    d_axe = d_axe - my
    return Smax*np.exp(-0.5*(d_axe / iy)**2)

# boucle pour retrouver les id_anneau qui ont suffisamment de points autour pour effectuer le calage
# Rq: df_calage est le tableau contenant id_anneau, id_capteur, d_axe et Se (s*)
id_anneau_a_caler = []
for id_anneau in df_calage.id_anneau.drop_duplicates().values:
    df = df_calage[(df_calage.id_anneau >= id_anneau - 20) &
                   (df_calage.id_anneau <= id_anneau + 20)]

    if (len(df) < 5 or len(df[df.d_axe.abs() <= 5]) < 1
                    or len(df[df.d_axe.abs() >= 15]) < 1):
        continue  # Pas assez de points
    id_anneau_a_caler.append(id_anneau)

# boucle pour caler l'equation du tassement transversal sur les mesures
for id_anneau in id_anneau_a_caler:
    df = df_calage[(df_calage["id_anneau"] >= id_anneau - 20) &
                   (df_calage["id_anneau"] <= id_anneau + 20)]
    X = df["d_axe"].values
    Y = df["Se"].values
    (Smax,iy, m) = curve_fit(S_trans, X, Y,
                             bounds = ([-20, 5, -5], [0, 35,5]),
                             p0 = [-2, 15, 0],
                             maxfev = 5000)
```

Script 5.4 Calage de l'équation transversale du tassement (Smax = s_{max}, d_{axe}, i_y, m_y)

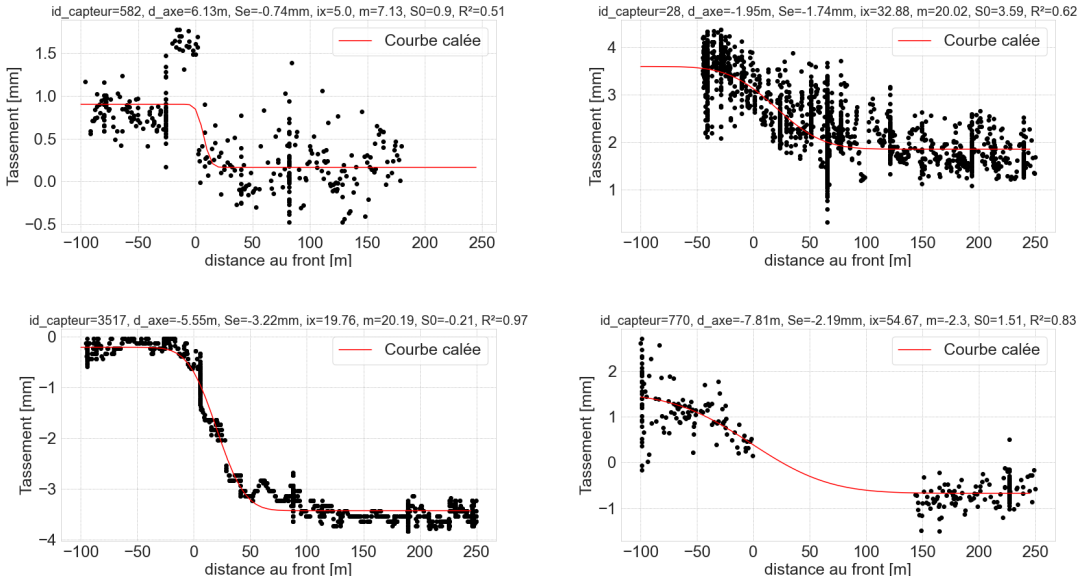


Figure 5.12. Exemples de capteurs retenus avec R^2 entre 0.5 et 1

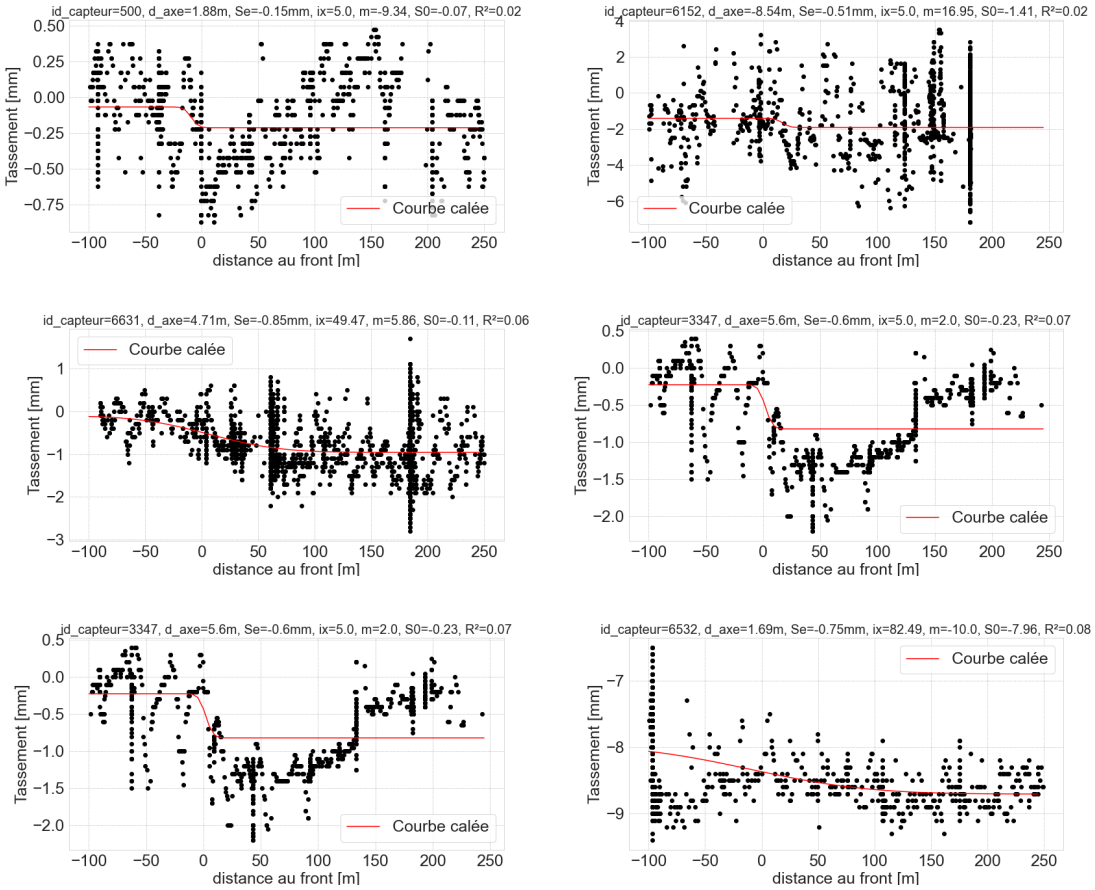


Figure 5.13. Exemples de capteurs retenus avec R^2 entre 0.02 et 0.5

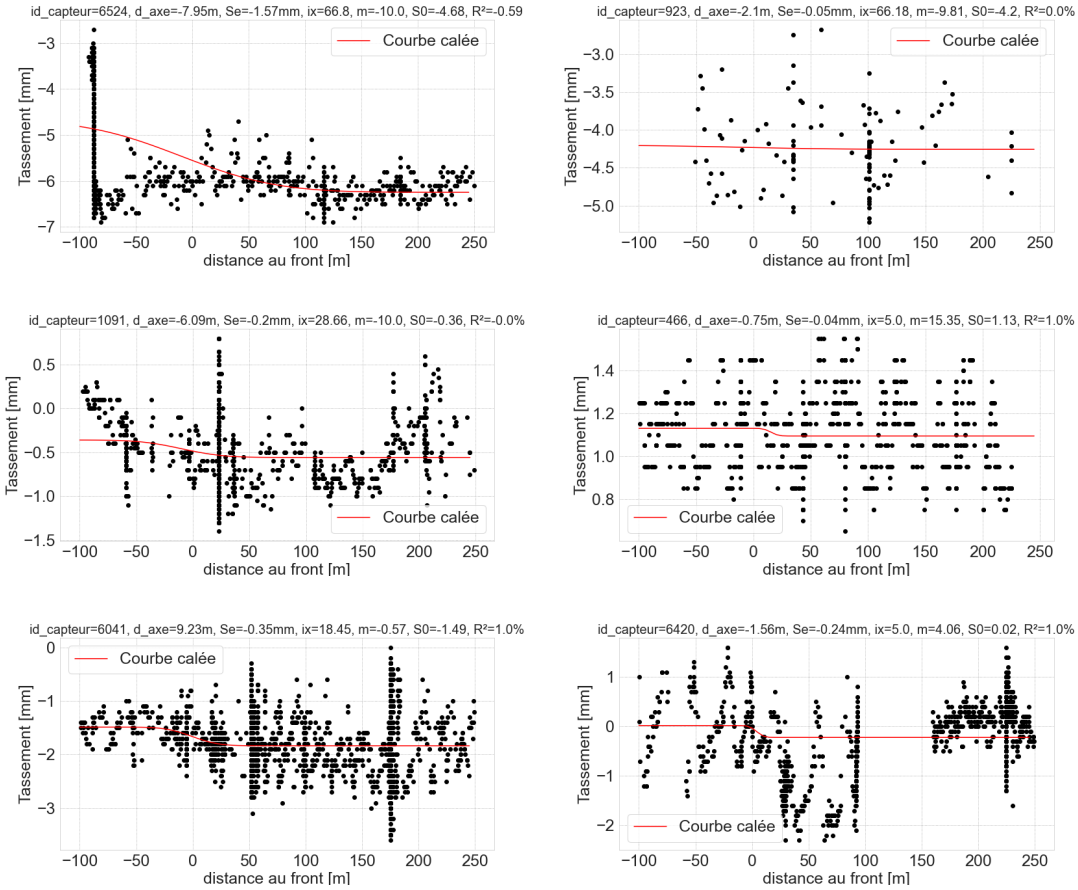


Figure 5.14. Exemples de capteurs non retenus ($R^2 < 0.02$)

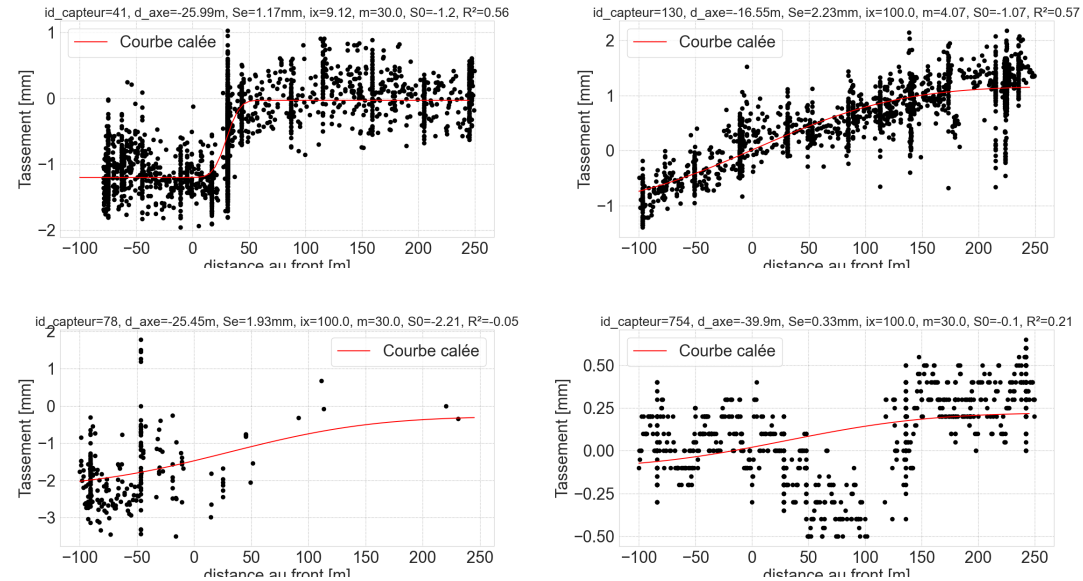


Figure 5.15. Exemples de résultats de calage de courbes de progression du tassement sur des mesures indiquant des soulèvements

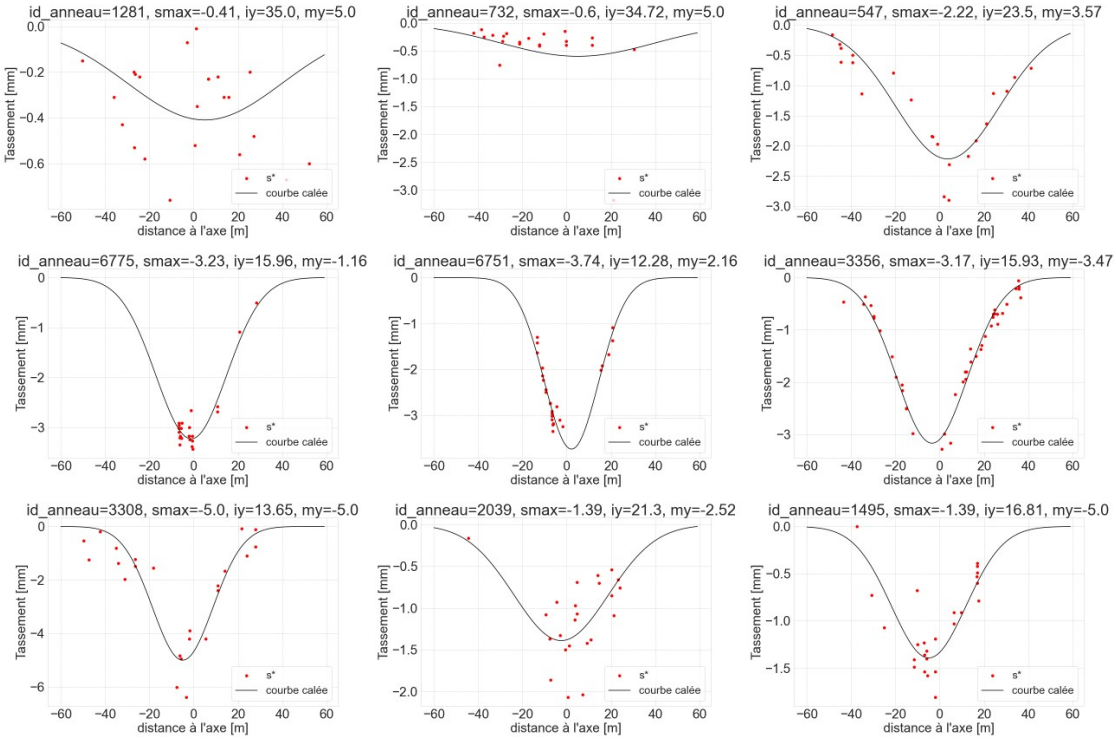


Figure 5.16. Quelques exemples de calage de l'équation transversale du tassement

5.3 Exploration et analyses statistiques des paramètres

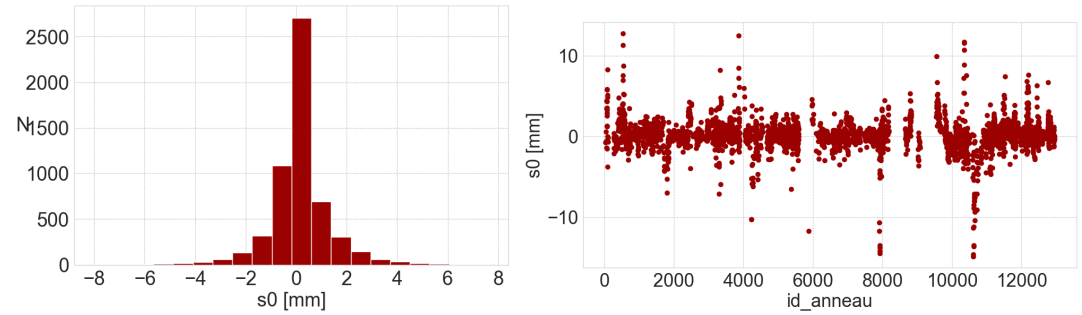
5.3.1 Exploration

Cette partie vient en continuité à l'analyse exploratoire des données. On y trouve les distributions des différents paramètres d'entrée, leur transformation si besoin ainsi que les études de corrélations entre les différents paramètres et le tassement.

Analyse du décalage s_0

Pour rappel, s_0 désigne la valeur de décalage de l'équation de progression du tassement. Ce paramètre varie majoritairement dans l'intervalle $[-4\text{ mm}\,;\,4\text{ mm}]$ (Figure 5.17a),à l'exception de 33 capteurs ayant des valeurs dans la plage $[-15\text{ mm}\,;\,15\text{ mm}]$ et 2 autres capteurs aux valeurs très en dehors de cette plage.

Les valeurs de s_0 en fonction de l'***id_anneau*** sont présentées dans la Figure 5.17b. On en conclut que les valeurs atypiques de s_0 se retrouvent majoritairement dans des zones concentrées (d'où les points sur des verticales) et qu'il n'y a pas de tendance générale qui montrerait des dérives. Il s'agit donc d'aberrations localisées qu'il n'est pas possible de corriger par le simple jeu de filtres ou d'offsets sectorisés.



(a) Distribution de s_0
35 valeurs supérieures à ± 8 mm ne sont pas visibles

(b) s_0 en fonction de la position représentée par id_anneau

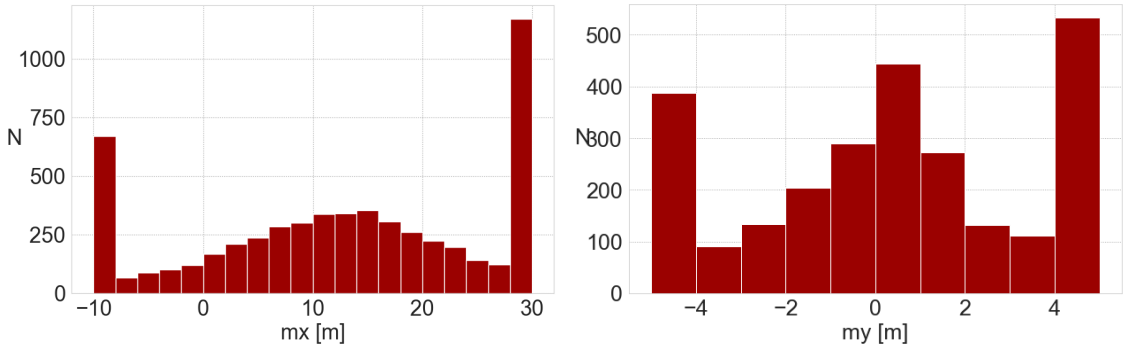
Figure 5.17. Analyses de s_0

Analyse de m_x et m_y

Pour rappel, m_y est le décalage par rapport à l'axe du tunnel où l'on observe s_{max} alors que m_x est le décalage de la courbe de tassement longitudinal qui rend compte du fait que le ratio entre le tassement au passage du front de taille et le tassement maximal observé en ce point (s^*) est variable.

Les valeurs de m_x présentent une distribution gaussienne avec une moyenne autour de 12 m (Figure 5.18a). Cela indique que, généralement, la moitié du s^* est atteinte à une distance d_{front} d'environ 12 m. Les valeurs de m_x montrent clairement un effet de bord aux bornes imposées lors du calage, soit -10 m et 30 m. Ces effets de bords impliquent que le calage n'est pas optimal mais il reste néanmoins exploitable. En effet, il convient de noter qu'il a été vérifié que les valeurs en dehors de la plage sont le cumul de valeurs distribuées de façon décroissante et monotone. Il n'y a donc pas d'autres modes cachés dans cette queue de distribution.

Quant aux valeurs de m_y, elles présentent également une distribution gaussienne centrée (Figure 5.18b), ce qui veut dire que majoritairement, le tassement maximal est bien observé directement au-dessus de l'axe du tunnel.



(a) Distribution de m_x

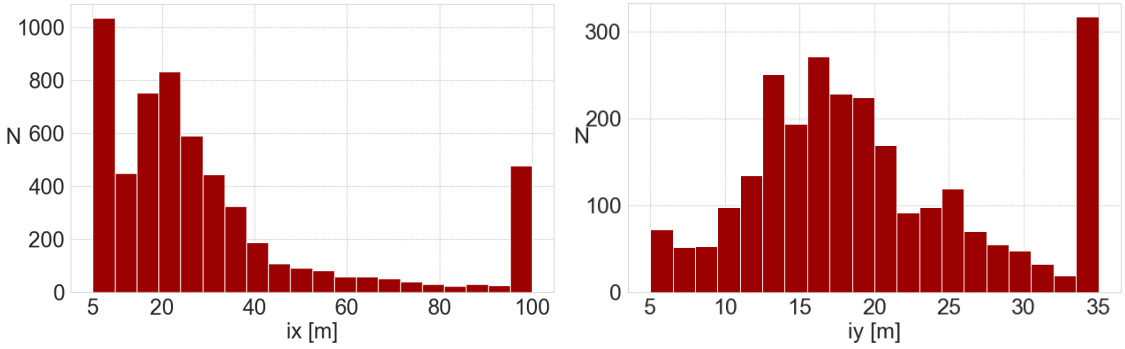
(b) Distribution de m_y

Figure 5.18. Distributions des paramètres m_x et m_y

Analyse de i_x et i_y

Pour rappel, i_y est la distance du point d'inflexion à l'axe de la courbe du tassement transversale alors que i_x est un facteur de forme qui rend compte de la position du point d'inflexion de la dérivée de cette gaussienne cumulée (et donc de la pente maximum de la courbe).

Les valeurs de i_y sont majoritairement réparties dans l'intervalle $[\ 5\ m\ ;\ 30\ m]$ avec un effet de bord clair sur la borne supérieure imposée qui est égale à 35 m (Figure 5.19b). En comparant les valeurs de i_y à celles de i_x, on remarque que les deux distributions sont plutôt similaires, à l'exception de la borne supérieure de i_x, qui était moins contrainte lors du calage, puisque limitée à 100 m. Compte tenu du faible nombre de valeurs entre 40 m et 100 m, on peut se dire que la limite supérieure de i_x a été mal choisie et devrait donc être limitée à une valeur autour de 40 m. Ce sujet fera l'objet d'une discussion à la fin de cette thèse, discussion portant plus généralement sur l'effet des bornes imposées aux paramètres calés sur la valeur de s^* (§ 7.2.2).



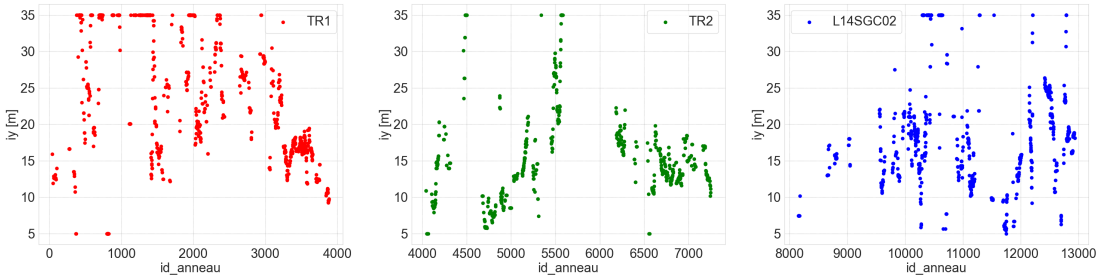
(a) Distribution de i_x (b) Distribution de i_y

Figure 5.19. Distributions des paramètres i_x et i_y

Une observation supplémentaire est faite sur la répartition des paramètres i_x et i_y en fonction de l'espace, représenté par ***id_anneau***. La Figure 5.20 montre que la répartition des valeurs de i_y suit des tendances variant selon les zones de creusement. Concernant les valeurs de i_x, il est difficile de tirer une conclusion claire en regardant tous les capteurs (représentés par ***id_capteur***) (Figure 5.21). On trace alors i_x en fonction de l'***id_anneau*** pour les capteurs à une distance à l'axe inférieure ou égale à 5 m (ce qui correspond à $D/2$) (Figure 5.22). On observe qu'il ya bien une certaine continuité des i_x sur le linéaire.

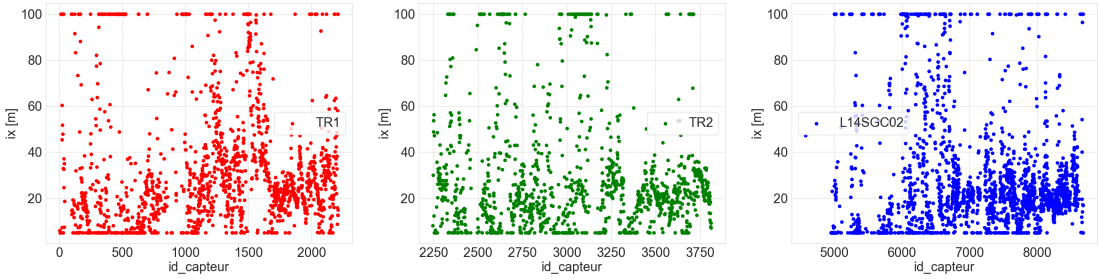
Analyse de s^*

Pour rappel, s^* est le tassement maximal observé par un capteur (point fixe de l'espace). Les valeurs de s^* obtenues à partir du calage de l'équation de progression du tassement sur les mesures de tassement indiquent qu'il y a des capteurs qui mesurent des soulèvement (s^* positifs, Figure 5.15). Ces capteurs sont considérés comme non-pertinents pour la



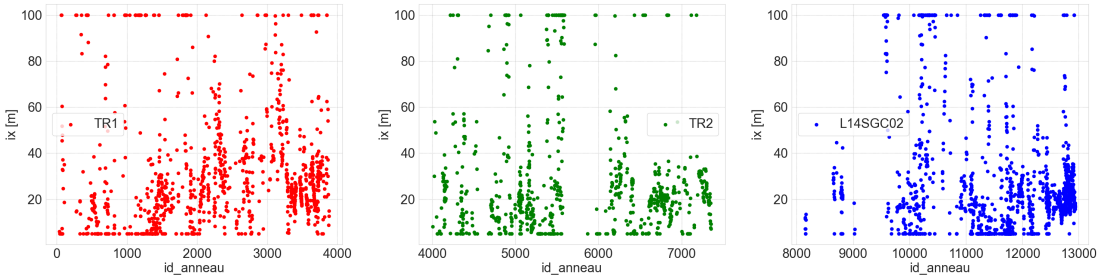
(a) TR1 (b) TR2 (c) L14S2

Figure 5.20. i_y en fonction de l'***id_anneau*** pour chaque troncon



(a) TR1 (b) TR2 (c) L14S2

Figure 5.21. i_x en fonction de l'***id_capteur*** pour chaque troncon



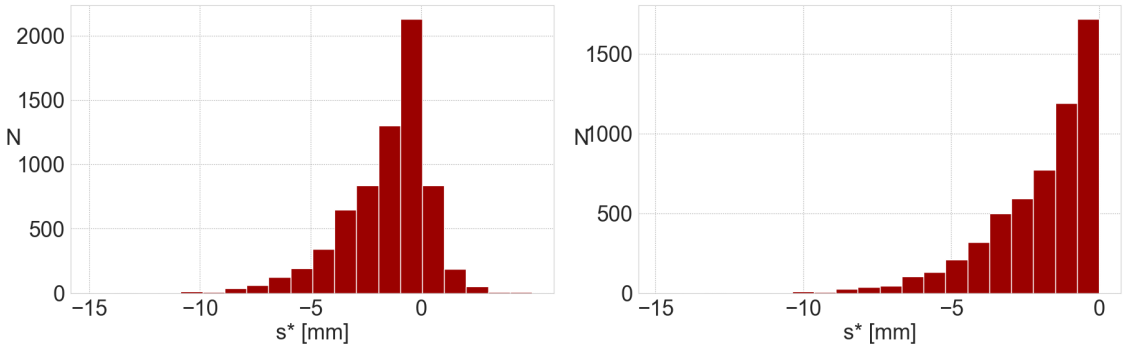
(a) TR1 (b) TR2 (c) L14S2

Figure 5.22. i_x en fonction de l'***id_anneau*** pour chaque tronçon (capteurs avec $d_{axe} \leq 5$ m)

prédiction des cuvettes de tassement et ne sont donc pas conservés dans la suite des études.

Les distributions du paramètre s^* issues du calage avant et après ce filtrage sur les valeurs positives de s^* sont présentées dans la Figure 5.23a et la Figure 5.23b. Les valeurs finales (filtrées) de s^* étudiées dans la suite varient majoritairement dans l'intervalle $[-10 \text{ mm} ; 0 \text{ mm}]$ hors valeurs atypiques qui sont très localisées mais hors distribution.

On trace également les valeurs de s^* en fonction de d_{axe} et des différents secteurs. Cette visualisation nous permet de distinguer d'un côté les secteurs qui ont le plus de mesures et d'un autre une allure des cuvettes de tassement dans chacune des zones avec une indication sur la valeur de s_{max} à laquelle il faut s'attendre. A titre d'exemple, on voit que les secteurs 20 et 21, qui correspondent aux ouvrages annexes de Jean Prouvé



(a) Distribution de s^* avant filtrage
8 valeurs inférieures à -15 mm ne sont pas visibles

(b) Distribution de s^*
5 valeurs inférieures à -15 mm ne sont pas visibles

Figure 5.23. Distributions du paramètre s^*

et République (200 m de longueur au total) n'ont pas suffisamment de valeur de s^* et par conséquent n'auront pas de valeurs de s_{max}. Cette information est confirmée par la distribution des ***id_anneau*** (Figure 5.25b) qui montrent un creux vers nomcolonneP-Kid_anneau $= 9000$ qui correspond au début de la L14S2. C'est effectivement une zone sur la L14S2 où il y a très peu de capteurs posés en surface (sud de la L14S2, Figure 5.7), ce qui justifie le faible nombre de s_{max}.
De plus, on observe que les s^* les plus importants, soit entre -12 mm et -15 mm, sont observés dans le secteur 1 (sans tenir compte du secteur 21 et de la valeur aberrante dans le secteur 11). Le secteur 1 correspond aux derniers 950 m du tronçon TR1, soit autour de la gare Fort d'Issy/ Vanves/ Clamart.

D'ailleurs, il est également possible de tracer la distribution de la distance des capteurs à l'axe du tunnel ainsi que sa répartition sur les trois tronçons (Figure 5.25). Ces figures montrent que la distribution des capteurs autour du tracé du tunnel est homogène, et cela quel que soit le tronçon, puisque la distribution de la distance à l'axe est normale centrée. On a donc des mesures de s^* en toute position par rapport à l'axe du tunnel.

Exploration de la progression du tassement

Après obtention des courbes calées de la progression du tassement avec l'avancement du tunnelier, il est possible d'effectuer des explorations statistiques sur les valeurs du tassement en fonction de la distance au front.

On s'intéresse tout d'abord à la valeur du ratio du tassement observé au front par rapport au tassement maximal s^*. Nous avons nommé cette valeur η, avec $\eta = s(d_{front} = 0)/s*$. On représente dans la Figure 5.26a une boîte à moustache des valeurs de η en fonction de la ligne de creusement pour les capteurs à une distance à l'axe du tunnel inférieure à 5 m ($D/2$). On trouve que les valeurs de η se concentrent majoritairement dans l'intervalle $[\,0.2\,;\,0.4]$. En d'autres termes, le tassement au front est généralement égal dans notre cas à 20 à 40% du tassement maximal final s^*. La Table 5.1 résume les

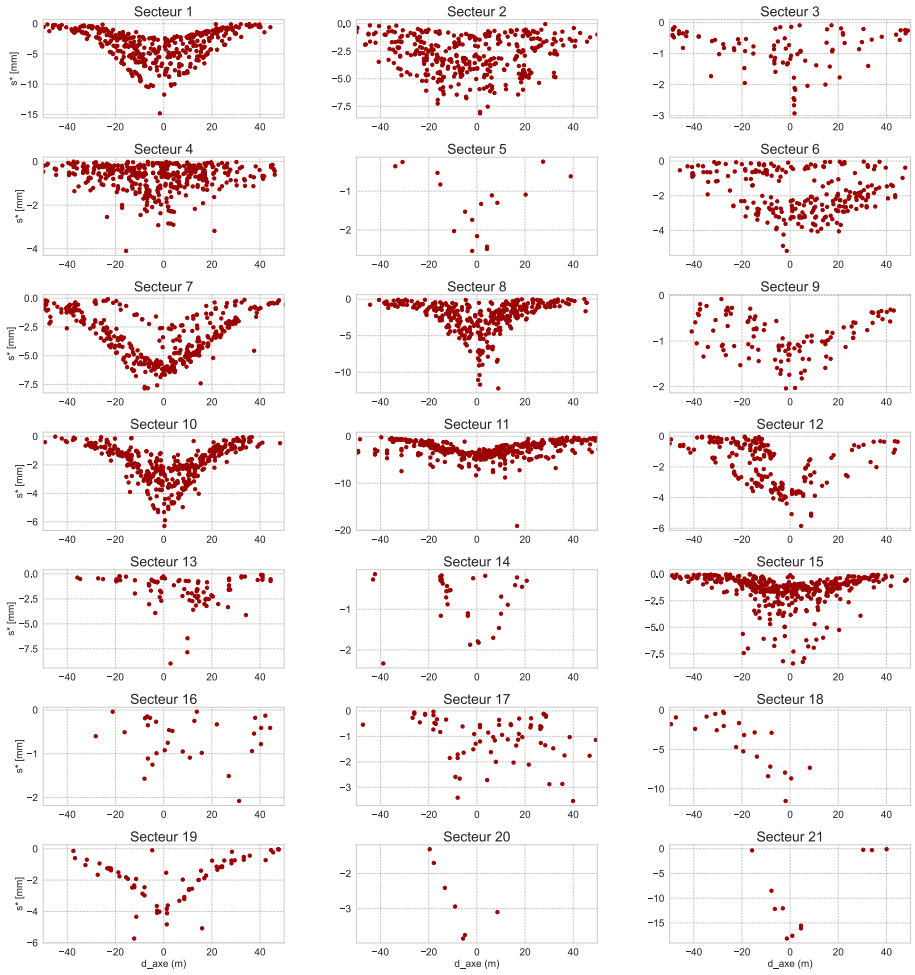


Figure 5.24. s^* en fonction de la distance du capteur à l'axe du tunnel et du secteur. Pour la liste des secteurs, se référer à la Table 4.1

tendances : la moyenne de η est de 0.33 pour les capteurs à des distances à l'axe arrivant jusqu'à 70 m.

Ensuite, on souhaite évaluer la distance au front du tunnel à partir de laquelle le tassement atteint 95% du tassement maximal s^*. Cette valeur est notée $d_{front\ s*}$. De même, on représente une boîte à moustache des valeurs de $d_{front\ s*}$ en fonction de la ligne de creusement. Le résultat, présenté dans la Figure 5.26b, montre que cette distance varie majoritairement entre 25 m et 75 m pour les capteurs à une distance à l'axe du tunnel inférieure à 5 m. La Table 5.1 indique que $d_{front\ s*}$ a pour moyenne 63 m, pour médiane 50 m et pour 1^{er} et $3^{ème}$ quartiles 33 m et 75 m, respectivement. Cela nous

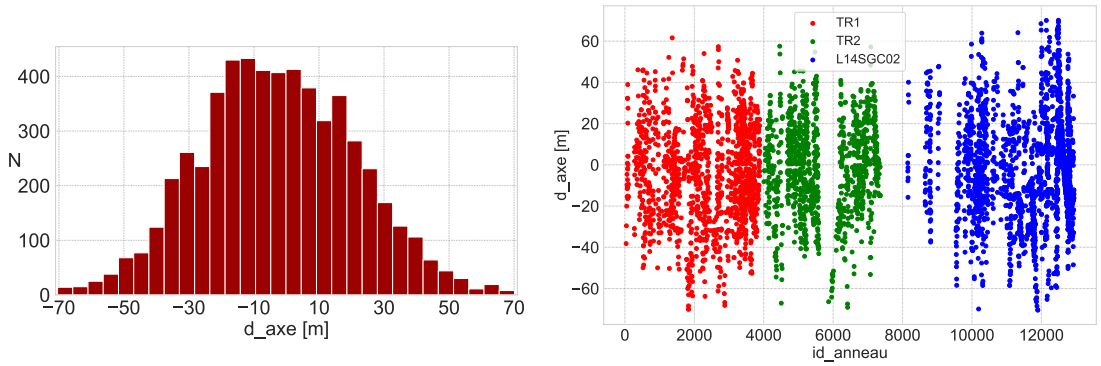


Figure 5.25. Distribution de la distance des capteurs à l'axe du tunnel (à noter que ce sont uniquement les capteurs avec un calage de l'équation de progression du tassement réussi)

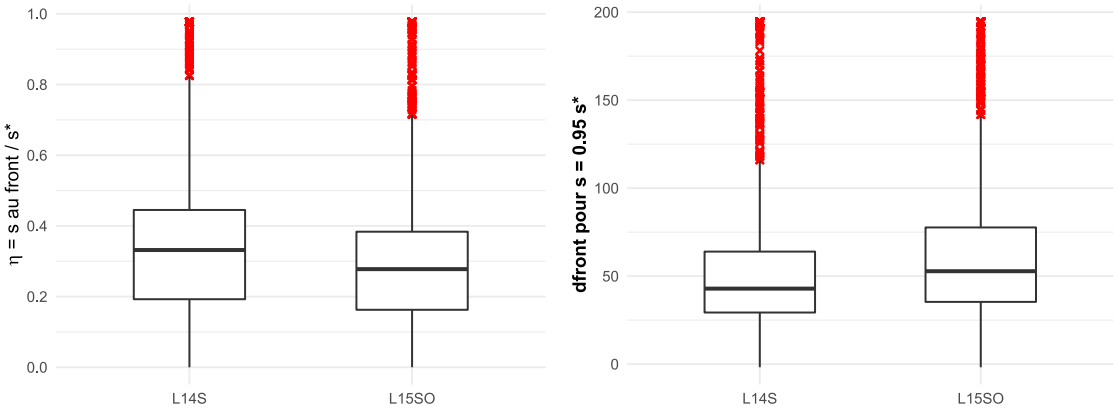
Table 5.1. Description statistique de η et $d_{front\ s*}$

	d_{axe} [m]	$\eta = s(d_{front} = 0)/s*$	$d_{front\ s*}$ [m]
nombre	5687	5687	5687
moyenne	-2.65	0.33	63.53
écart-type	23.75	0.22	49.50
min	-70.40	0.00	-1.78
25%	-19.12	0.19	33.55
50%	-3.13	0.32	49.89
75%	14.27	0.42	75.15
max	70.12	0.98	194.49

permet donc de conclure que la longueur de la zone de transition entre le moment où un capteur n'est pas affecté par le creusement et le moment où il atteint son tassement maximal est typiquement de l'ordre d'une centaine de mètres.
On peut donc simplifier le problème d'apprentissage en ignorant cette zone complexe, où les valeurs de tassement varient significativement. Nous y reviendrons dans la prochaine partie (§ 7.2).

Analyse de s_{max}

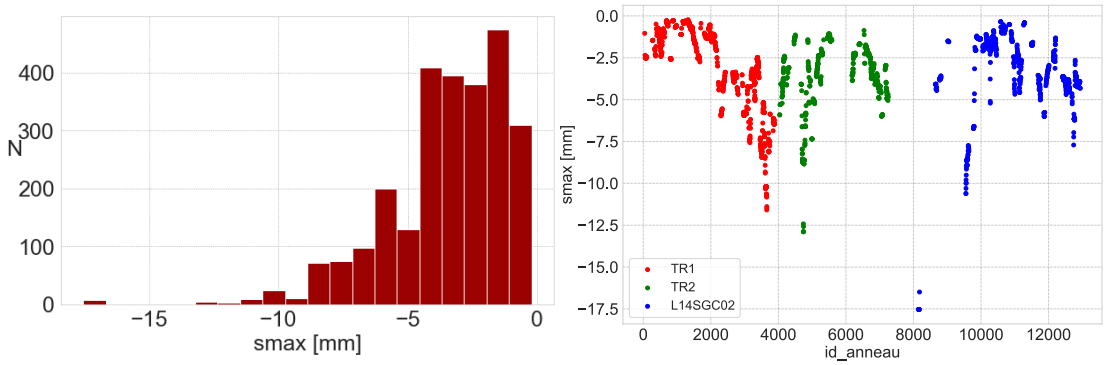
Le paramètre suivant à analyser est le tassement maximal observé à l'axe du tunnel s_{max}. Il convient de rappeler qu'une des conclusions obtenues à partir de la Figure 5.3 est que les valeurs extrêmes de s_{max} sont observées dans les zones avec les plus faibles couvertures. La distribution des valeurs de s_{max} ainsi que leur répartition sur le tracé du tunnel sont présentés dans la Figure 5.27. Les résultats montrent que s_{max} varie dans typiquement dans l'intervalle $[-12\text{ mm}; 0\text{ mm}]$. De plus, on remarque que les tassements les plus forts sont observés sur la L14S2, pour les ***id_anneau*** autour de 8000. Cette zone est à proximité de l'ouvrage annexe Jean Prouvé. On peut faire l'hypothèse que les travaux de cet ouvrage ont influencé la valeur de s_{max}. La description statistique de s_{max} est présentée dans la Table 5.2.



(a) Ratio du tassement au front par rapport au tassement maximal s^* (η)

(b) Distance au front pour laquelle le tassement est égal à 95% du tassement maximal s^* ($d_{front\ s*}$)

Figure 5.26. Boîte à moustache en fonction de la ligne pour des capteurs avec $d_{axe} \leq 5$ m



(a) Distribution de s_{max}

(b) Répartition de s_{max} sur le tracé

Figure 5.27. Analyses des valeurs de s_{max}

Distribution des caractéristiques

L'étape suivante consiste à tracer la distribution des caractéristiques choisies précédemment (§ 5.1.2, Figure 5.28) ainsi qu'à établir leurs mesures statistiques (Table 5.2). Il convient de mentionner à cette étape que les algorithmes d'apprentissage automatique choisis (algorithmes à base d'arbres de décision, § 6.1) ne sont pas dépendants de la distribution des paramètres. Il n'est donc pas nécessaire d'effectuer des normalisations des distributions à ce stade.

Nombre d'anneaux posés par jour

Pour le retour d'expérience, nous avons effectué des statistiques sur le nombre d'anneau posés par jour sur les trois tronçon TR1, TR2 et L14S2. Ces valeurs sont représentées dans la Figure 5.29. Le nombre maximal d'anneaux posés par jour est de 17, soit un creusement de 34 m (un anneau a une longueur de 2 m). On observe dans les trois cas une progression

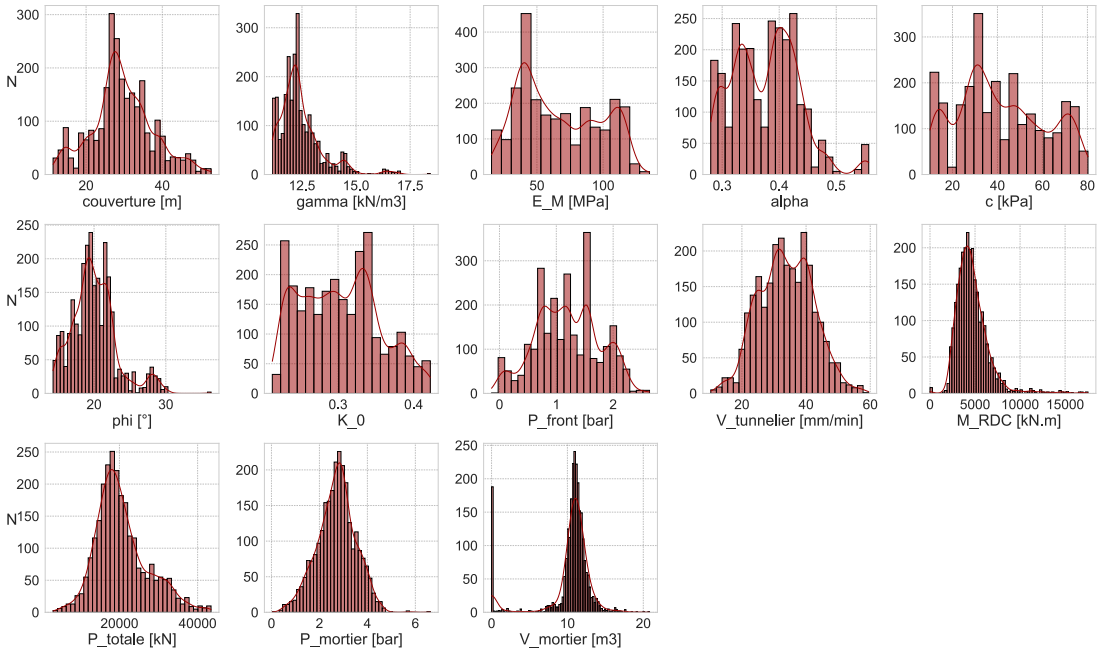


Figure 5.28. Distributions des paramètres

	s_{max}	$V_{tunnelier}$ [mm/-min]	M_{RDC} [kN.m]	P_{front} [bar]	$P_{mortier}$ [bar]	$V_{mortier}$ [m^3]	P_{totale} [kN]
nombre	2590.0	2590.0	2590.0	2590.0	2590.0	2590.0	2590.0
moyenne	-3.5	33.7	4687.1	1.2	2.7	10.3	20599.4
écart-type	2.4	8.4	1852.5	0.6	0.8	3.3	6726.5
min	-17.5	10.3	0.0	-0.1	0.0	0.0	3013.8
25%	-4.5	27.5	3538.4	0.8	2.2	10.2	16146.1
50%	-3.2	33.6	4387.1	1.2	2.7	11.0	19266.9
75%	-1.7	39.8	5409.8	1.6	3.2	11.7	23899.4
max	-0.2	59.5	17540.6	2.6	6.6	20.8	43442.2

	C [m]	γ [kN/m^3]	E_M [MPa]	α	c [kPa]	φ [°]	K_0
nombre	2590.0	2590.0	2590.0	2590.0	2590.0	2590.0	2590.0
moyenne	29.5	12.4	66.6	0.4	41.4	20.0	0.3
écart-type	8.0	1.0	30.2	0.1	19.2	3.1	0.1
min	11.3	11.1	15.7	0.3	10.0	14.3	0.2
25%	25.2	11.8	40.3	0.3	28.4	18.2	0.3
50%	28.7	12.2	60.5	0.4	38.3	19.7	0.3
75%	34.4	12.8	92.5	0.4	55.6	21.7	0.3
max	52.8	18.4	135.1	0.6	80.2	36.3	0.4

Table 5.2. Description statistique des variables

systématique en début de creusement, qui correspond à l'apprentissage des opérateurs et, au tout début, au montage de la machine. Au niveau du TR2, on observe une période creuse début 2020 qui correspond au coincement de la machine après le passage de la Bièvre.

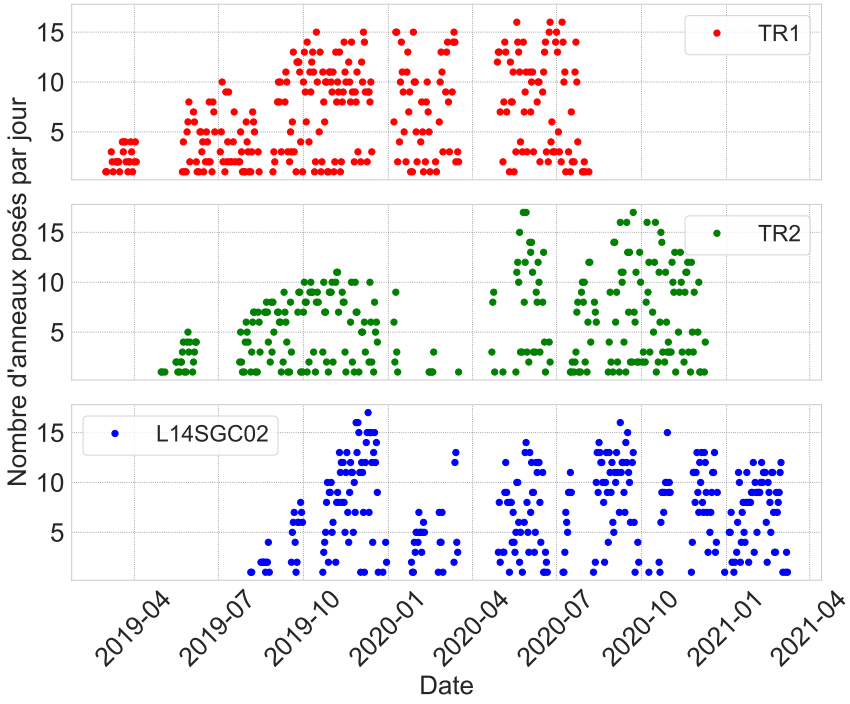


Figure 5.29. Nombre d'anneaux posés par jour en fonction du tronçon

5.3.2 Analyses bivariées

L'étape suivante consiste à étudier la corrélation des paramètres entre eux ainsi que celle des paramètres avec la variable cible à l'aide du coefficient de Pearson R (Équation 2.1). Le coefficient de corrélation de Pearson varie de -1 à 1 et indique une corrélation parfaite négative (R=-1) ou positive (R=1) entre les deux variables étudiées.

Tout d'abord, une carte de chaleur (heatmap) est tracée en prenant s_{max} comme variable cible. Les résultats, présentés dans la Figure 5.30, indiquent que les paramètres les plus corrélés entre eux sont c [kPa] et E_M [MPa] avec $R = 0.86$, puis K_0 et α avec $R = 0.84$ et ensuite φ [°] et γ [kN/m^3] avec $R = 0.74$. Ces corrélations ne sont pas analysées en recherchant un sens physique puisque ce sont des paramètres de sols combinés.

La corrélation la plus forte avec s_{max} est celle de K_0 ($R = 0.56$), tandis que la corrélation la plus faible est celle avec M_{RDC} [kN.m] ($R = 0.07$). Nous avons donc choisi de ne pas prendre en compte M_{RDC} [kN.m] en tant que caractéristique pour l'entraînement des modèles d'apprentissage automatique. Nous remarquons sur la Figure 5.31 que c'est K_0 qui figure en première place pour s_{max} et en 2ème place pour s^* après

d_{axe}. Les paramètres suivants dans le classement sont c [kPa], α et E_M [MPa] pour s_{max} et α et c [kPa] pour s^*. On remarque que l'influence des paramètres n'est pas la même sur les deux variables cibles en question, notamment pour la couverture du tunnel.

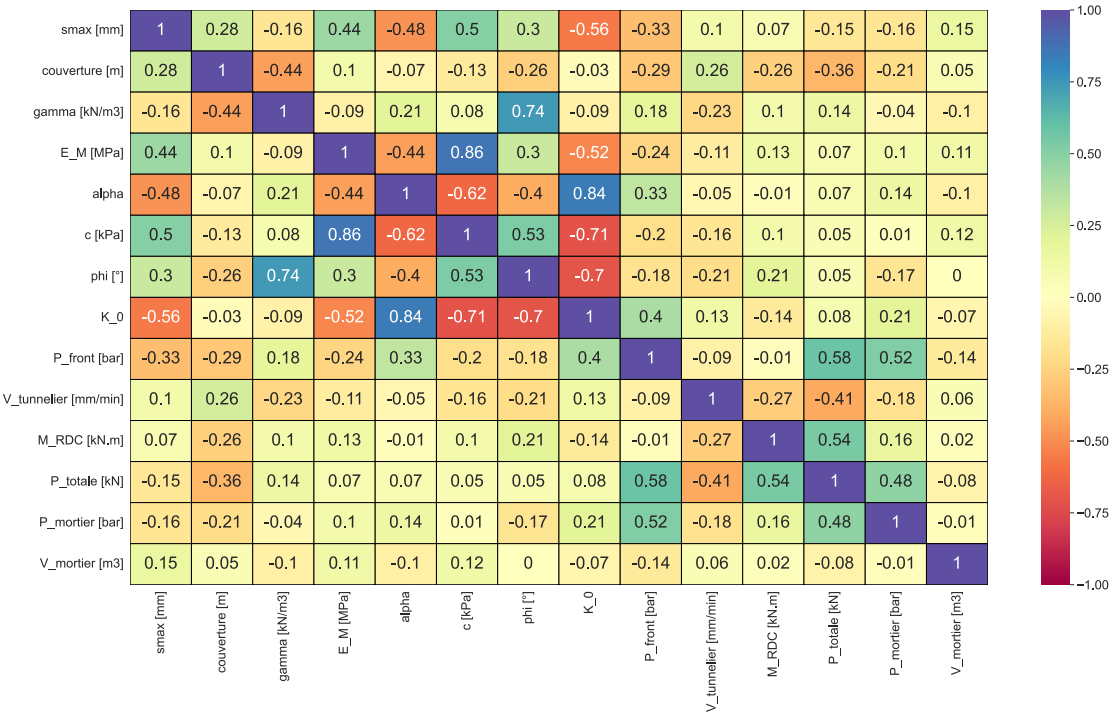


Figure 5.30. Carte de chaleur (Heatmap) des paramètres ayant une influence sur le tassement et le tassement maximal à l'axe s_{max}

Conclusion

Le présent chapitre était composé de trois sections distinctes.
Tout d'abord, une mise au point sur les objectifs des travaux est effectuée afin de sélectionner les variables cibles et les caractéristiques pertinentes. Nous avons présenté des techniques d'extraction de caractéristiques, notamment le calcul de nouveaux paramètres tels que la distance d'un capteur à l'axe du tunnel (d_{axe}) et la réduction de la dimension des paramètres géologiques et géotechniques.

Les variables cibles choisies sont le tassement maximal observé à l'axe du tunnel (s_{max}) ainsi que le tassement maximal à une distance de l'axe du tunnel (s^*). La deuxième partie présente alors des techniques supplémentaires de nettoyage des mesures de tassement telles que l'algorithme d'apprentissage automatique des forêts d'isolation (IF). Nous avons ensuite calé l'équation de progression du tassement sur les mesures nettoyées des capteurs pour obtenir s^*, puis calé l'équation du tassement transversal sur les valeurs de s^* ainsi obtenues pour déterminer s_{max}.

Une fois que les paramètres nécessaires à notre étude ont été obtenus, nous avons pu explorer la variabilité de ces paramètres afin de mieux les connaître et déceler d'éven-

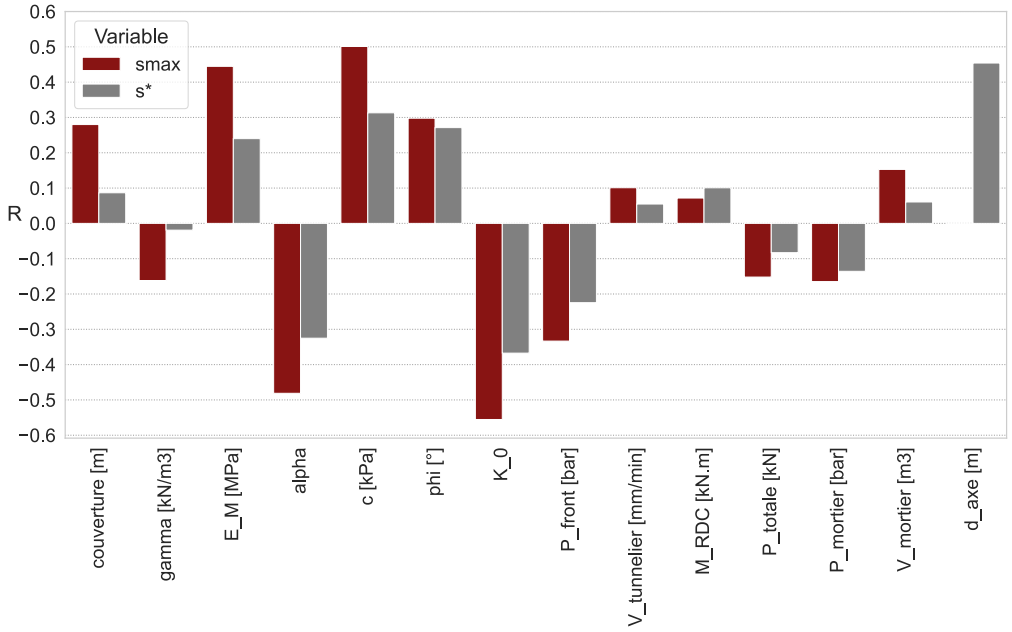


Figure 5.31. Coefficient de corrélation de Pearson R de s_{max} et de s^* avec les paramètres

tuelles erreurs lors des processus de transformation et de calage. Des analyses statistiques ont été menées pour mieux comprendre les données et déduire les corrélations entre les différents paramètres et les variables cibles. Cela nous a permis d'affiner la sélection des caractéristiques à utiliser pour entraîner les algorithmes d'apprentissage automatique.

En somme, ce chapitre nous a permis d'approfondir notre compréhension des données, d'extraire les caractéristiques pertinentes et de calibrer les équations nécessaires à notre étude. Ces résultats seront utilisés pour la modélisation et la prévision des tassements dans le chapitre suivant à l'aide d'algorithmes d'apprentissage automatique tels que les forêts aléatoires.

Conclusion

La présente partie était composée de deux chapitres.

Le Chapitre 4 avait pour objectif de décrire les lignes 14 Sud et 15 Sud-Ouest ainsi que la collecte, le nettoyage et le stockage des données. A ce stade, nous avons enfin une base de données opérationnelle qui servira pour les études effectuées dans le cadre de cette thèse ou encore pour des études ultérieures.

Le Chapitre 5 a servi d'abord à effectuer une mise au point sur les objectifs des travaux afin de sélectionner les variables cibles et les caractéristiques pertinentes. Une réduction des dimensions des paramètres géologiques et géotechniques est présentée ainsi que des techniques supplémentaires de nettoyage des mesures de tassement telles que l'algorithme d'apprentissage automatique des forêts d'isolation (IF). Par la suite, on a effectué un calage des équations de progression du tassement et du tassement transversal afin d'obtenir les variables cibles (s^* et s_{max}). A partir de ce stade, on a réalisé des études statistiques pour retrouver les paramètres les plus influants sur les valeurs du tassement.

La partie suivante sera dédiée à l'utilisation d'algorithmes d'apprentissage automatique pour la prévision des tassements. Des recommandations sur la construction d'une base de données seront également présentées.

Partie III

Prédiction des tassements à l'aide d'outils d'apprentissage automatique

Introduction

Grâce à l'ingénierie des données du Grand Paris Express, nous avons à notre disposition une grande quantité de données sur laquelle il est possible d'effectuer la prévision des tassements.

Nous tenons tout d'abord à préciser un point de sémantique. Dans cette thèse, nous avons en effet privilégié l'utilisation du terme « prévision », alors même qu'il est d'usage dans la terminologie du Machine Learning d'utiliser le terme de « prédiction ». En effet, si l'on s'en réfère à la définition du verbe prédire, l'acception la plus courante est celle d'« annoncer d'avance ce qui doit arriver, par intuition, raisonnement ou conjecture, par une inspiration prétendument surnaturelle » (Larousse, 2023a). Même s'il existe bien dans la même source un sens plus neutre à ce terme, la prédiction dans son sens littéral contient donc une connotation antinomique avec l'aspect scientifique du travail qui cherche à être accompli dans cette thèse. Ce terme est susceptible d'être mal compris par un public d'ingénieurs. A sa décharge, le terme dérive de l'anglais, où la connotation n'est pas si prononcée. En revanche, le verbe prévoir (forecast) renvoie au fait de « penser, d'après certaines données, qu'un fait futur est très probable » (Larousse, 2023b). L'emploi du mot « prévision » nous semble donc plus approprié, d'autant qu'il nous permet de dresser un parallèle avec la météorologie, cette dernière étant bien une **science** qui a pour objet l'étude des phénomènes atmosphériques et leur « prévision ». Quels que soient les modèles et algorithmes utilisés en météo, leurs fondements sont scientifiques, et les prévisions sont aujourd'hui de plus en plus fiables et indispensables dans beaucoup de cas d'usages (trafic aérien, agriculture etc...). Notre étude a bien, de la même façon, vocation à justifier les estimations qui sont établies, et à les rendre utiles en pratique.

Cette partie est dédiée à de nombreuses approches de prévision du tassement maximal observé en surface suite au creusement des tunnels. D'abord, nous adoptons une approche simple qui consiste à diviser aléatoirement les données. La régularisation et l'optimisation des algorithmes d'apprentissage automatique à base d'arbres de décision sont présentées également. Ensuite, on prend en compte la spatialité du problème : on prévoit le tassement à l'avant du front, en entraînant les modèles uniquement sur les données à l'arrière de celui-ci. Enfin, on teste un cas pratique qui tient compte de l'aspect spatio-temporel du tassement en tenant compte de la position du front et de la disponibilité des données à une date donnée.

Expérimentations avec division aléatoire des données

6

Introduction

Dans le chapitre précédent, nous avons effectué des analyses exploratoires afin de mieux comprendre la grande quantité de données à notre disposition et de détecter l'influence des différents paramètres sur les tassements. Ces études ont permis d'obtenir les variables cibles et de sélectionner les caractéristiques à utiliser pour entraîner les algorithmes d'apprentissage automatique. A ce stade, il est donc possible de procéder à des études de prévision de tassement à l'aide de ces algorithmes.

Ce chapitre s'intéresse à une division aléatoire des données, c'est-à-dire une distribution aléatoire des données d'entraînement et de test sur le tracé du tunnel. Il est essentiel de commencer par une telle approche simple pour de nombreuses raisons. Tout d'abord, on sait qu'aujourd'hui les algorithmes d'apprentissage automatique sont performants en interpolation, mais pas nécessairement en extrapolation (§ 3.4.2). Une division aléatoire est une approche qui teste la capacité de généralisation d'un modèle sans vérifier ses performances en extrapolation. Ensuite, à partir de cette approche, il est possible de développer une méthodologie claire de validation des modèles obtenus. Cela permet de comparer la performance des algorithmes choisis afin de sélectionner le (ou les) meilleur(s) pour une application approfondie. Enfin, il est possible à ce stade de régulariser les modèles afin d'obtenir des résultats plus fiables vis-à-vis du sur-apprentissage.

Deux exercices sont proposés dans ce chapitre : la prévision du tassement maximal au droit de l'axe du tunnel (s_{max}) et la prévision du tassement maximal à n'importe quelle distance de l'axe du tunnel (s^*). En premier lieu, on décrit la méthodologie adoptée pour les deux exercices. Ensuite, on présente la prévision du tassement s_{max} et on termine par la prévision du tassement s^*.

6.1 Définitions et méthodologie

Cette partie a pour objectif d'expliciter les choix effectués pour résoudre les deux problèmes de prévision évoqués ci-dessus : choix des variables cibles, des caractéristiques et des algorithmes d'apprentissage automatique à tester. Ensuite, on présente la normalisation des données et on discute les méthodes possibles de division des données. Enfin, on décrit la méthode adoptée pour valider les modèles obtenus.

Table 6.1. Caractéristiques utilisées pour la prévision de s_{max}

Catégorie	Caractéristique
Géométrie	Couverture C [m]
Pilotage du tunnelier	Vitesse d'avancement $V_{tunnelier}$ [mm/min] Pression au front P_{front} [bar] Poussée totale du tunnelier P_{totale} [kN] Pression de mortier injecté $P_{mortier}$ [bar] Quantité de mortier injecté $V_{mortier}$ [m^3]
Géologie et géotechnique (paramètres « combinés »)	Poids volumique γ [kN/m^3] Module pressiométrique de Ménard E_M [MPa] Coefficient rhéologique α Cohésion c [kPa] Angle de frottement φ [°] Coefficient de pression des terres au repos K_0

6.1.1 Généralités

Variable cible et caractéristiques

On cherche à prévoir le tassement maximal au droit de l'axe du tunnel noté s_{max} ou bien le tassement maximal à une distance de l'axe du tunnel s^*. Pour cela, on utilise les valeurs de s_{max} et s^* calées précédemment (§ 5.2.3).

Il convient de noter que, par la suite, nous n'avons pas exclu les zones qui pourraient potentiellement combiner des déformations issues d'autres phénomènes, comme les zones proches des gares ou des ouvrages annexes. La zone d'influence de ces ouvrages reste en effet relativement restreinte par rapport au tunnel (quelques dizaines de mètres tout au plus).

Les caractéristiques sont sélectionnées selon l'état de l'art (§ 3.3.1) ainsi que les analyses statistiques effectuées sur nos ensembles de données (§ 5.3.2). Les caractéristiques choisies pour la prévision de s_{max} sont présentées dans la Table 6.1. Pour la prévision de s^*, les mêmes caractéristiques sont utilisées avec en plus la distance à l'axe du tunnel (en valeur absolue).

Algorithmes d'apprentissage automatique

Afin de choisir les algorithmes d'apprentissage automatique à tester, il faut tout d'abord catégoriser l'apprentissage (§ 2.2.2). Dans le cadre de ce travail, l'apprentissage se caractérise de la façon suivante :

– *Apprentissage sur la base de modèles :*
les algorithmes d'apprentissage automatique doivent être basés sur l'optimisation d'un modèle à travers la détection des tendances dans les données d'apprentissage.

– *Apprentissage par lot statique de données :*
les données sont collectées au préalable de l'entraînement (pas de flux de données).

– *Apprentissage supervisé :*
la sortie cible (s_{max} ou s^*) est une information connue. Nous avons donc en notre possession des données étiquetées qui serviront à l'apprentissage de l'algorithme.

– *Apprentissage pour la régression :*
la sortie cible est une valeur numérique continue.

En se basant sur les catégories d'apprentissage présentées ci-dessus, les algorithmes sélectionnés sont : Linear Regression LR, Support Vector Machine Regressor SVM, Back-Propagation Neural Network BPNN, Decision Tree Regressor DT, Random Forest Regressor RF, et Extreme Gradient Boosting Machine XGBoost.

Spécificités d'architecture des réseaux de neurones

Avant d'entraîner un réseau de neurones BPNN, il faut choisir son architecture. A partir de retours d'expériences et en procédant par essais / erreurs, nous avons sélectionné l'architecture suivante : une couche d'entrée (input layer) avec un nombre de neurones égal au nombre de caractéristiques, quatre couches cachées (hidden layers) avec respectivement 29, 16, 16, et 8 neurones et enfin une couche de sortie (output layer) avec un seul neurone (un seul paramètre de sortie). De plus, on choisit la fonction d'activation tangente hyperbolique (*tanh*) avec *Adams* comme optimiseur (optimizer, algorithme de descente de gradient) et un taux d'apprentissage (learning rate) de 0.001. La même architecture est utilisée pour la prévision de s_{max} et s^*.

```
# {python}
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras import backend as K
# erreur choisie: RMSE
def cust_RMSE(y_true, y_pred):
    return K.sqrt(K.mean(K.square(y_pred - y_true)))
# Architecture du reseau
np.random.seed(0)
acti = "tanh" # fonction d'activation
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(units = 29, activation = acti,
    input_shape = (X_train.shape[1],)))
model.add(tf.keras.layers.Dense(units = 16, activation = acti))
model.add(tf.keras.layers.Dropout(0.1))
model.add(tf.keras.layers.Dense(units = 16, activation = acti))
model.add(tf.keras.layers.Dense(units = 8, activation = acti))
model.add(tf.keras.layers.Dense(units = 1, activation = 'linear'))
model.compile(optimizer=tf.keras.optimizers.Adam(0.001), loss =
    cust_RMSE)
```

Script 6.1 Architecture du réseau de neurones BPNN

Il convient de noter que l'application des réseaux de neurones a fait l'objet d'un stage au sein de Setec terrasol, mais n'a pas été concluant (Lebdaoui, 2022). Néanmoins, nous présentons quelques résultats dans la suite pour comparaison.

6.1.2 Préparation des données

Mise à l'échelle des caractéristiques

Les algorithmes de SVM et BPNN sont sensibles à l'échelle des données d'entrée. Il convient donc de normaliser les données, ce qui signifie de mettre toutes les caractéristiques sur la même échelle en les transformant en une plage standard (§ 3.3.3). Cela permet au modèle d'effectuer une analyse plus précise et plus cohérente des données, sans être influencé par l'échelle ou la mesure des différentes caractéristiques. Dans ce cas, on décide d'appliquer la standardisation sur les données à l'aide de la fonction *StandardScaler* de la librairie *scikit-learn*. L'influence de cette normalisation sur les résultats de SVM sont discutés dans ce qui suit.

```python
# {python}
from sklearn.preprocessing import StandardScaler

X_train_sc = StandardScaler().fit_transform(X_train)
X_test_sc = StandardScaler().fit_transform(X_test)

y_train_sc = StandardScaler().fit_transform(y_train.reshape(-1, 1))
y_test_sc = StandardScaler().fit_transform(y_test.reshape(-1, 1))
```

Script 6.2 Standardisation des données

Division des données

L'ensemble des données est divisé en un ensemble d'apprentissage et un autre de test. La fonction *train_test_split* de la librairie *scikit-learn* permet de diviser les données de différentes manières.

```python
# {python} {exemple sur smax}
from sklearn.model_selection import train_test_split
# choix des caracteristiques
features = ['couverture [m]',
            'gamma [kN/m3]', 'E_M [MPa]', 'c [kPa]', 'phi', 'K_0',
            'P_front [bar]', 'V_tunnelier [mm/min]', 'M_RDC [kN.m]',
            'P_totale [kN]', 'P_mortier [bar]', 'V_mortier [m3]']
# choix du pourcentage des donnees de test
# division des donnees: 20% pour le test
X_train, X_test, y_train, y_test =
    train_test_split(all_data[features], all_data['smax [mm]'],
                     test_size = 0.2, random_state = 0)
```

Script 6.3 Division des données avec la fonction *train_test_split*

En plus des caractéristiques et de la variable cible, la fonction *train_test_split* prend également les arguments suivants :

- *test_size* : représente la portion de l'ensemble de test.
- *shuffle*[True/False] : contrôle si les données sont mélangées avant leur division (split).
- *stratify* : si *shuffle=True*, on peut choisir de diviser les données de manière stratifiée. Cela signifie que le processus de division prend en compte la répartition des classes de la variable cible.
  Il convient d'attirer l'attention sur le fait que l'usage du terme *classe* implique la présence d'une variable catégorielle.

Pour comparer l'effet des paramètres *shuffle* et *stratify* sur la répartition des données d'apprentissage et de test, les divisions suivantes des données sont testées avec l'ensemble de données de s_{max} :

```
# {python} {exemple sur smax}
# pas de shuffle
X_train, X_test, y_train, y_test =
    train_test_split(all_data[features],
                     all_data["smax [mm]"],
                     test_size=0.2,
                     shuffle = False, random_state=0)

# shuffle
X_train, X_test, y_train, y_test =
    train_test_split(all_data[features],
                     all_data["smax [mm]"],
                     test_size=0.2,
                     shuffle = True, random_state=0)

# shuffle et stratify (creation d'une variable categorielle de smax)
all_data = all_data.assign(smax_cat =
    pd.cut(all_data['smax [mm]'],
           bins=[-20, -15, -10, -5, 0],
           labels=['[-20, -15]', '[-15, -10]',
                   '[-10, -5]', '[-5, 0]']))
X_train, X_test, y_train, y_test =
    train_test_split(all_data[features],
                     all_data["smax [mm]"],
                     test_size=0.2,
                     shuffle = True, random_state=0,
                     stratify = all_data["smax_cat"])
```

Script 6.4 Comparaison de l'effet des paramètres *shuffle* et *stratify* de la fonction *train_test_split*

Les résultats de ce test sont représentés sur la Figure 6.1. On rappelle qu'on n'a pas de mesures de s_{max} à chaque mètre, ce qui explique les vides dans la répartition des données. Pour mieux comprendre la division stratifiée des données, on observe la répartition des valeurs de s_{max} dans les ensembles d'apprentissage et de test (avec et sans stratification)

dans la Figure 6.2. On remarque que la répartition des classes de la version catégorielle de s_{max} est respectée avec une division stratifiée des données. Cela permet de s'assurer que la division des données est correctement répartie entre les ensembles d'apprentissage et de test, ce qui garantit de meilleurs résultats puisqu'on n'aura pas de valeurs en test qui n'existent pas en apprentissage. Toutefois, nous avons choisi de ne pas utiliser cette approche de stratification car les distributions de nos variables cibles sont plutôt bien réparties.

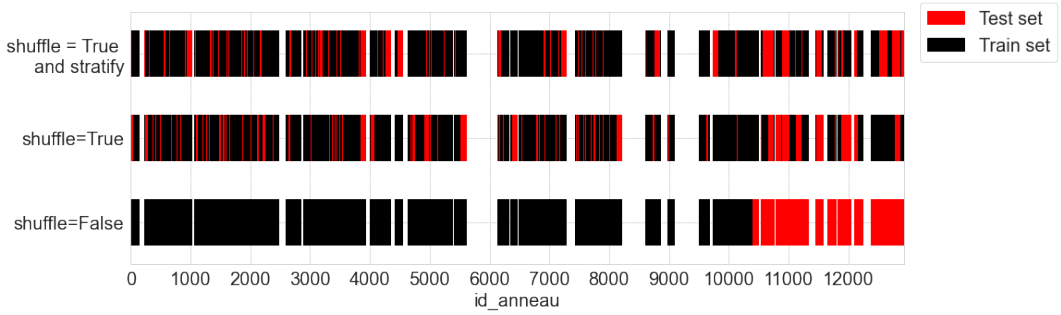


Figure 6.1. Observation de l'effet des paramètres *shuffle* et *stratify* de la fonction *train_test_split*

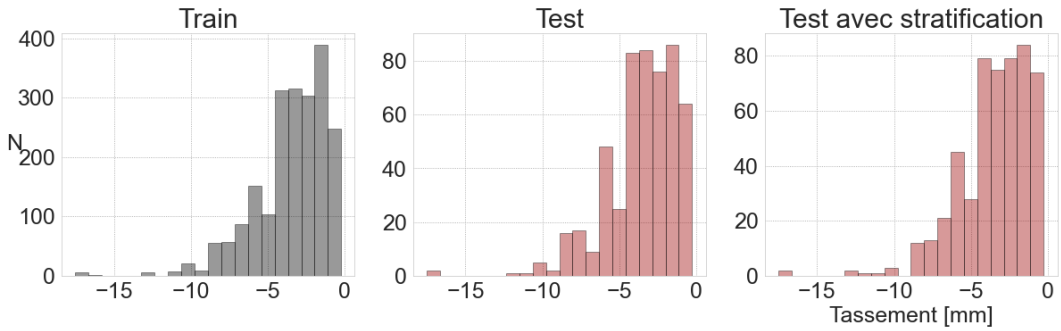


Figure 6.2. Répartition des valeurs de s_{max} dans l'ensemble d'apprentissage et l'ensemble de test, avec comparaison entre une division des données stratifiées ou non

6.1.3 Validation des algorithmes

Après la division des données en ensembles d'apprentissage et de test, il convient d'optimiser la performance des modèles sur l'ensemble d'apprentissage. Pour cela, une méthode unique est développée pour comparer les différents modèles testés. On choisit d'effectuer une validation croisée (cross-validation, § 2.2.1) avec 5 plis (folds) pour observer la performance sur l'ensemble de validation et détecter la présence de sur-apprentissage (overfitting) des modèles. Pour évaluer les résultats de cette validation croisée, on trace les courbes d'apprentissage (learning curves). Ces courbes représentent l'évolution du taux d'erreur des ensembles d'apprentissage (courbe d'apprentissage) et de validation (courbe de validation) en fonction de la taille du jeu de données d'apprentissage. L'indicateur d'erreur choisi est le coefficient de détermination R^2 (Équation 2.2). Nous nous intéressons également à la racine carrée de l'erreur quadratique moyenne RMSE (Équation 2.6) pour

évaluer la performance sur l'ensemble de test. Dans la suite, nous désignons les courbes d'apprentissage par le terme en anglais « learning curves » pour éviter toute confusion entre *les courbes d'apprentissage* (courbe d'apprentissage et courbe de validation) et *la courbe d'apprentissage* (score sur l'ensemble d'apprentissage).

```python
# {python}
def evaluation(model, X_train, y_train):
    # N: taille des ensembles d'apprentissage
    N, train_score, val_score =
        learning_curve(model, X_train, y_train,
                       shuffle = True, random_state = 0,
                       cv = 5, scoring = "r2",
                       train_sizes = np.linspace(0.1, 1.0, 10))

    # plot
    plt.plot(N, train_score.mean(axis=1), label = "train score")
    plt.plot(N, val_score.mean(axis=1), label = "val score")
```

Script 6.5 Learning curves

6.2 Prévision du tassement maximal à l'axe

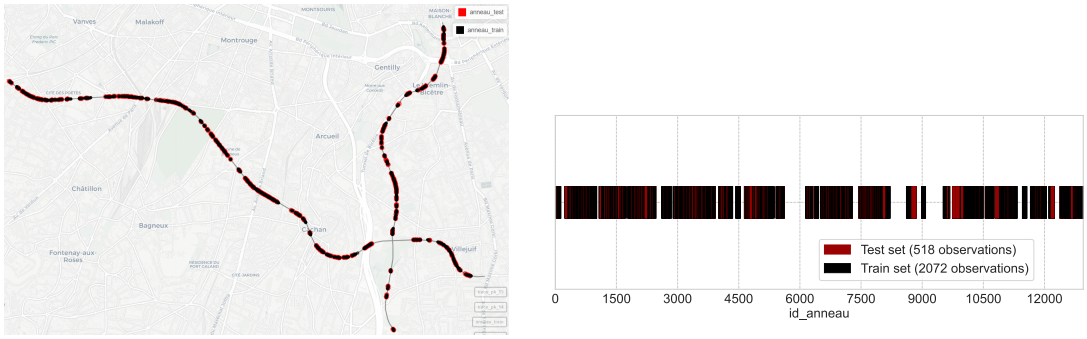
Nous présentons ici les résultats des modèles de prévision de s_{max} pour une division aléatoire des données, avec différents ratios entre données d'apprentissage et données de test. Ensuite, on détaille la méthode de régularisation et d'optimisation des algorithmes à base d'arbres de décision (DT, RF et XGBoost).

6.2.1 Apprentissage avec 80% des données

La division choisie est aléatoire avec mélange des données et sans stratification. Le jeu de données contient 2592 observations qui sont divisés en 80% pour l'ensemble d'apprentissage et 20% pour l'ensemble de test, soit 2072 et 518 observations respectivement. La répartition des deux ensembles selon l'***id_anneau*** ainsi que sur les tracés des deux lignes est présenté dans la Figure 6.3. La distribution des paramètres dans les ensembles d'apprentissage et de test est montrée dans la Figure 6.4.

Les résultats des différents algorithmes sont regroupés dans la Figure 6.12. On remarque que les modèles LR et SVM donnent des résultats médiocres avec des R^2 respectifs de 0.44 et 0.07. Néanmoins, suite à la standardisation des données, l'algorithme SVM montre une très bonne performance (Figure 6.12k), soit un score R^2 de 0.91 sur les données d'apprentissage et 0.84 sur les données de test. De plus, on remarque sur les learning curves que ce modèle n'est pas en sur-apprentissage (la courbe d'apprentissage ne sature pas à un R^2 de 1) et est capable de généraliser (la courbe de validation tend vers la courbe d'apprentissage avec l'ajout de données).

Les modèles basés sur les arbres de décision (DT, RF et XGBoost) ont des R^2 d'apprentissage et de test proches de 1. Cependant, ces modèles ont clairement fait du



(a) (b)

Figure 6.3. Répartition des données sur le tracé du tunnel avec une division de 20% de l'ensemble des données pour le test

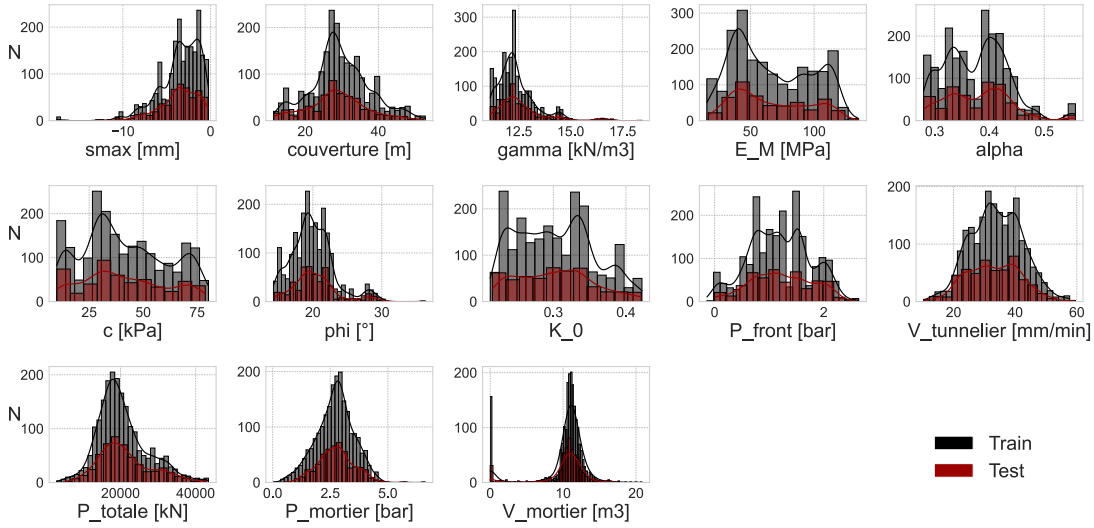


Figure 6.4. Distribution des caractéristiques avec une division de 20% de l'ensemble des données pour le test

sur-apprentissage puisque les learning curves montrent des courbes d'apprentissage avec un R^2 de 1 pour un nombre très faible de données. Il faut donc régulariser ces modèles, c'est-à-dire optimiser les hyperparamètres pour éviter le sur-apprentissage.

Concernant les réseaux de neurones, les résultats présentés montrent une bonne performance du modèle obtenu, soit un R^2 de 0.89 sur les données de test. Ce résultat est inférieur à celui de RF et XGBoost. Néanmoins, les learning curves des réseaux de neurones montrent que le modèle n'est pas en sur-apprentissage puisque la courbe de validation s'approche de la courbe d'apprentissage avec l'augmentation du nombre de cycles (epoch) sans pour autant la croiser.

En comparant les learning curves des algorithmes SVM, DT, RF et XGBoost, on remarque que DT a besoin d'un nombre de données plus grand que celui dont ont besoin RF et XGBoost pour atteindre de bonnes performances. Par exemple, pour 600 observations, SVM et DT enregistrent des R^2 autour de 0.8 sur les données de validation alors que RF et XGBoost arrivent à des R^2 de 0.9. On en conclut d'abord que les algorithmes basés sur

l'assemblage de modèles (ensemble methods) sont capables d'une généralisation bien plus remarquable que les modèles simples, et ensuite que ces algorithmes n'ont pas besoin d'une grande quantité de données pour fournir des modèles performants. Par conséquent, il convient de relancer les modèles avec un ensemble d'apprentissage plus petit.

6.2.2 Apprentissage avec 30% des données

Dans ce qui suit, on choisit de prendre un ensemble de test avec 70% du jeu de données, soit 1813 observations pour le test et 777 observations pour l'apprentissage. La répartition des deux ensembles selon l'***id_anneau*** ainsi que sur les tracés des deux lignes est présenté dans la Figure 6.5. La distribution des paramètres dans l'ensemble d'apprentissage et de test est montrée dans la Figure 6.6.

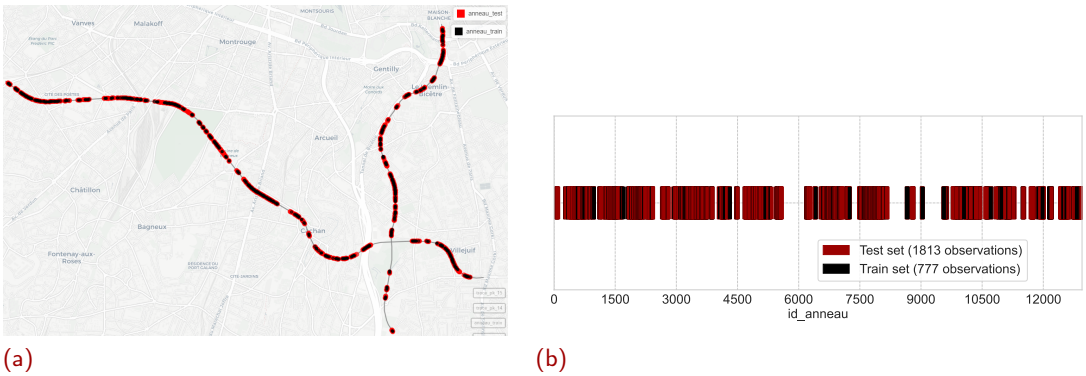


Figure 6.5. Répartition des données sur le tracé du tunnel avec une division de 70% de l'ensemble des données pour le test

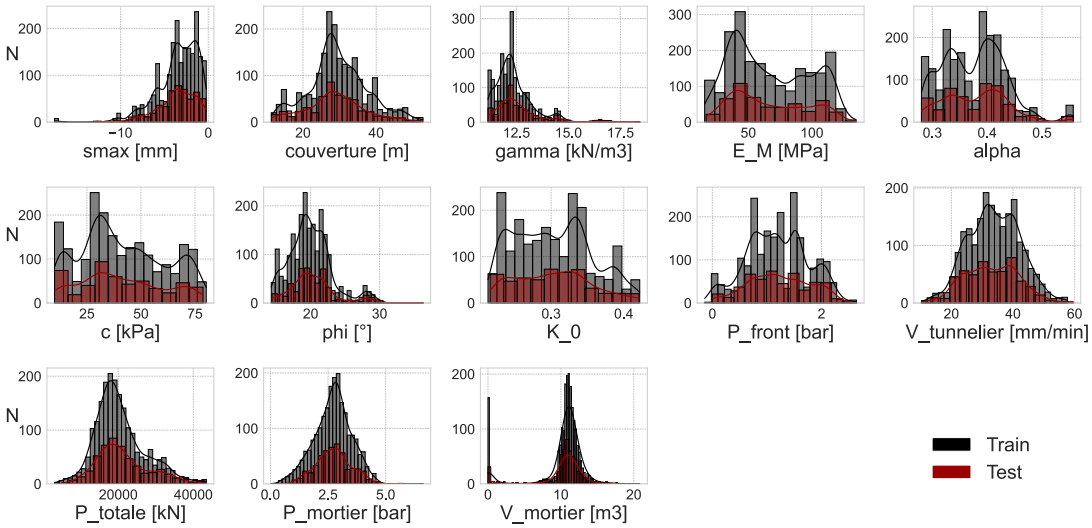


Figure 6.6. Distribution des caractéristiques avec une division de 70% de l'ensemble des données pour le test

Les résultats des différents algorithmes sont regroupés dans la Figure 6.13. Ces résultats confirment que 30% des données, soit 777 observations, semblent être une quantité suffisante pour entraîner les algorithmes DT, RF, XGBoost, SVM et BPNN. En effet, pour ces algorithmes, le score R^2 sur l'ensemble de test est respectivement de 0.82, 0.91, 0.93, 0.81 et 0.87. On conclut donc que l'algorithme de XGBoost semble être le plus performant, suivi de RF, ANN, DT et SVM. Toutefois, les modèles à base d'arbres continuent à sur-ajuster sur les données d'apprentissage puisque les courbes d'apprentissage ont un R^2 de 1. Il faut donc procéder à la régularisation des modèles à base d'arbres, c'est-à-dire à l'optimisation des hyperparamètres (§ 2.2.1). Cela n'est pas le cas de l'algorithme SVM puisque, d'un côté, la courbe d'apprentissage est croissante sans dépasser un R^2 de 0.9 et, d'un autre côté, la courbe de validation est croissante et arrive presque à des résultats identiques à ceux de la courbe d'apprentissage. L'algorithme SVM n'est pas conservé dans la suite car il a besoin de plus de données que les autres modèles pour être aussi performant.

Concernant les BPNN, on voit que le modèle commence à sur-ajuster sur les données d'apprentissage à partir d'environ 50 cycles. Nous avons donc arrêté l'entraînement à 80 cycles et regardé alors les résultats sur l'ensemble de test. On obtient un R^2 de 0.84 (contre 0.87 précédemment). Les réseaux de neurones ne seront pas retenus pour la suite de cette étude vu que leur optimisation est plus complexe que celle des modèles à base d'arbres et que ces derniers ont fait preuve de leur haute performance pour ce cas d'usage.

Dans ce qui suit, on choisit de régulariser tout d'abord l'algorithme DT qui est le plus simple afin de valider la méthodologie d'optimisation. Ensuite, on optimisera les algorithmes RF et XGBoost. A noter que les algorithmes seront entraînés avec 30% des données.

6.2.3 Régularisation et optimisation des hyperparamètres

Importance des caractéristiques

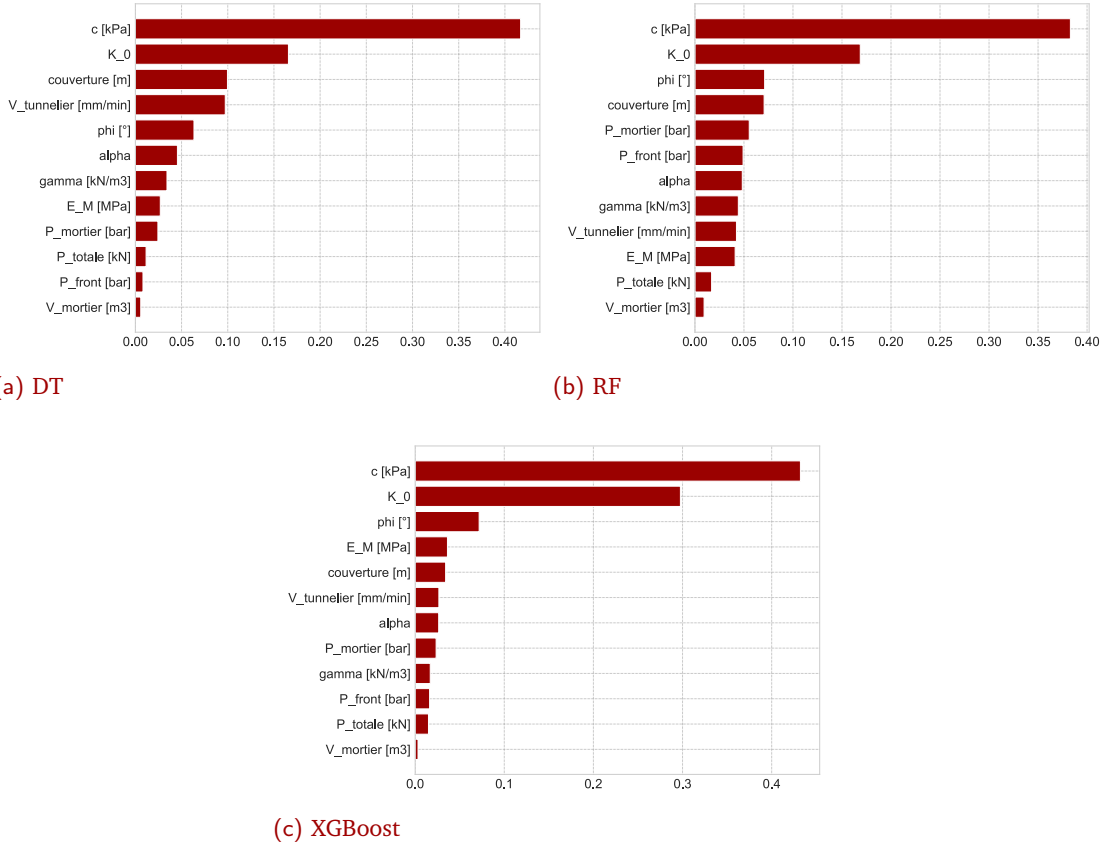
Avant de se lancer dans la régularisation des hyperparamètres des différents algorithmes, il convient d'optimiser la sélection des caractéristiques à travers le calcul de l'influence de chacune sur le modèle obtenu (Feature Importance). Pour les algorithmes à base d'arbres de décision tels que DT, RF et XGBoost, l'influence des caractéristiques est donnée par l'attribut *feature_importances_* inclus dans les algorithmes.

```
1 # {python}
2 def feature_importance_DT(model_DT, features):
3     # ici model_DT peut aussi etre RF ou XGBoost
4     plt.figure(figsize=(12,8))
5     plt.yticks(fontsize = 25)
6 
7     sorted_idx = model_DT.feature_importances_.argsort()
```

```
8      plt.barh(np.array(features)[sorted_idx], model_DT.
       feature_importances_[sorted_idx], color = "#9b0100")
```

Script 6.6 Fonction pour tracer l'importance des caractéristiques en ordre croissant

La Figure 6.7 montre que les deux caractéristiques $V_{mortier}$ [m^3] et P_{totale} [kN] n'ont quasiment pas d'influence sur les résultats des modèles des trois algorithmes. Ce résultat n'est pas surprenant puisque les études statistiques, spécifiquement le calcul du coefficient de corrélation de Pearson R entre s_{max} et les paramètres d'entrées (Figure 5.31), montre que les deux caractéristiques P_{totale} [kN] et $V_{mortier}$ [m^3] ont, respectivement, une corrélation de -0.15 et 0.15 avec s_{max}. Par conséquent, ces deux paramètres ne sont plus sélectionnés parmi les caractéristiques dans la suite.



(a) DT (b) RF

(c) XGBoost

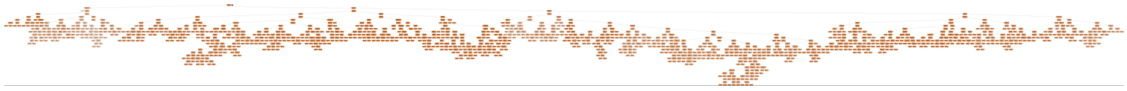
Figure 6.7. Importance des caractéristiques dans les modèle de DT, RF et XGBoost (entraînement avec 30% des données)

Les résultats du changement de caractéristiques est présenté dans la Figure 6.14. L'optimisation des caractéristiques a amélioré les résultats du modèle de DT avec un R^2 qui passe de 0.82 à 0.86 et un RMSE qui diminue de 1 à 0.88. Cependant, les algorithmes RF et XGBoost semblent être insensibles à ce changement de caractéristiques puisque les résultats restent quasiment les mêmes en termes de R^2 et RMSE. Il convient de noter également que l'importance des caractéristiques varie légèrement après modification des

caractéristiques. De plus, après cette modification, les trois algorithmes montrent que les trois caractéristiques ayant la plus grande importance par rapport à la valeur finale de s_{max} sont : c [kPa], K_0 et φ [°].

Optimisation de Decision Tree

La régularisation d'un arbre de décision (DT) peut se faire en limitant sa profondeur. Cet hyperparamètre est représenté par l'argument *max_depth* de la fonction *DecisionTreeRegressor*. Si cet hyperparamètre n'est pas spécifié, la valeur par défaut est *None*, ce qui signifie qu'il n'y a aucune limite sur la profondeur et donc l'arbre « grandit » autant que nécessaire pour trouver le meilleur modèle possible. En effet, si on observe l'arbre obtenu par le modèle sans régularisation, on retrouve un arbre très profond ce qui confirme encore une fois le sur-apprentissage du modèle (Figure 6.8).



(a) Arbre de décision généré par l'algorithme DT

(b) Arbre de décision généré par l'algorithme DT après régularisation des hyperparamètres

Figure 6.8. Arbre de décision généré par DT (entraînement avec 30% des données)

Afin de trouver la valeur optimale d'un hyperparamètre, on calcule le biais et la variance du modèle (§ 2.2.1). La valeur optimale de l'hyperparamètre est celle qui minimise ces deux erreurs (Figure 2.5).

La fonction *bias_variance_decomp* de la librairie *mlxtend* est utilisée pour effectuer une analyse de décomposition biais-variance pour un algorithme donné.

```python
# {python}
mse, bias, var = [], [], []
max_profondeur = range(1, 20, 2)
for i in max_profondeur:
    model = DT(random_state=0,  max_depth= i)
    _mse, _bias, _var =
        bias_variance_decomp(model, X_train, y_train, X_test, y_test,
                             loss='mse', num_rounds=10, random_seed=1)
    mse.append(_mse)
    bias.append(_bias)
    var.append(_var)
```

Script 6.7 Calcul du biais et de la variance

Les résultats de ce calcul (Figure 6.9) indiquent que la valeur optimale de la profondeur de DT est de l'ordre de 9. On remarque sur cette figure qu'à partir d'un *max_depth* de 11, l'erreur semble être stabilisée. On peut en déduire que, avec le nouveau choix de caractéristiques, l'arbre de décision se stabilise à une profondeur d'environ 11.

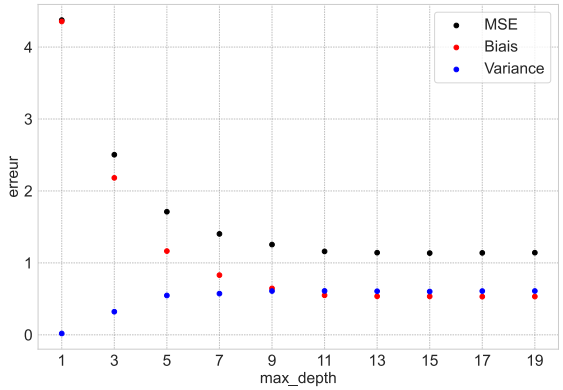


Figure 6.9. Calcul du biais et de la variance en variant l'hyperparamètre *max_depth* de DT

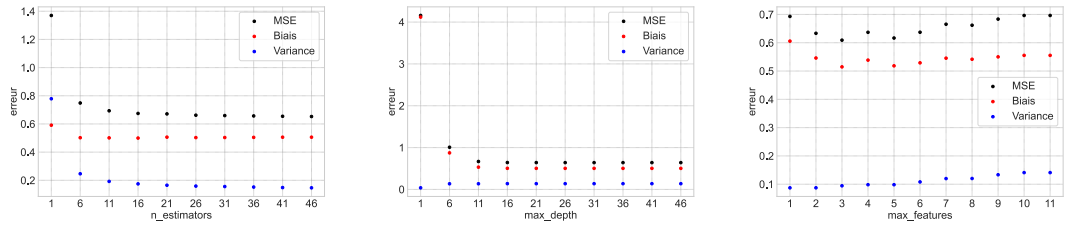
Les prédictions sont reprises avec le modèle régularisé de DT (*max_depth* = 9) et les résultats sont montrés dans la Figure 6.15a. Après régularisation, on obtient un modèle avec une performance légèrement inférieure (R^2 de 0.85 contre 0.86 auparavant) mais qui n'est plus en sur-apprentissage. En effet, le R^2 d'apprentissage n'atteint plus la valeur de 1 et cela peu importe le nombre de données. On remarque sur la courbe de validation que le modèle augmente considérablement sa performance avec le nombre de données. On en conclut que l'algorithme DT donne de meilleurs résultats avec plus de données pour l'entraînement.

Optimisation de Random Forest

Pour régulariser les forêts aléatoires, on choisit d'optimiser les hyperparamètres suivants :

- *n_estimators* : nombre d'arbres dans la forêt. La valeur par défaut est 100. D'une manière générale, un nombre plus grand d'arbres dans la forêt limite le sur-apprentissage mais augmente le temps de calcul.
- *max_depth* : profondeur maximale des arbres dans la forêt (estimateurs). La valeur par défaut est *None*, indiquant qu'il n'y a pas de limite.
- *max_features* : nombre de caractéristiques à prendre en compte lors de la recherche de la meilleure division d'un nœud. La valeur par défaut est 1. Il faut privilégier des estimateurs plus diversifiés et donc limiter la valeur de *max_features*. En effet, plus cette valeur est faible, plus la réduction de la variance est importante, mais le biais augmente dans le même temps (Sckit-learn, 2023). Pour une valeur de départ, on peut tester une valeur entre 30 et 50% du nombre de caractéristiques.

Selon les résultats observés dans la Figure 6.10, on trouve que les plages de valeurs optimales des hyperparamètres de RF sont : *n_estimators* supérieur à 16, *max_depth* entre 6 et 11 et *max_features* entre 3 et 6.



(a) n_estimators (b) max_depth (c) max_features

Figure 6.10. Biais et variance de RF en variant *n_estimators*, *max_depth* et *max_features*

Par la suite, on effectue une recherche aléatoire dans la plage des hyperparamètres retrouvés en utilisant la fonction *RandomizedSearchCV* de la librairie *scikit-learn*. Il convient de rappeler qu'il est recommandé d'avoir un grand nombre d'estimateurs, c'est pourquoi on propose une plage de valeurs de *n_estimators* entre le minimum obtenu (16) et 150.

```
# {python}
hyperparameters_RF = {'n_estimators': [15, 25, 35, 50, 75, 100, 150],
                      'max_features': range(3,7),
                      'max_depth': range(6,12)}
model = RandomForestRegressor(random_state=0)
rf_search = RandomizedSearchCV(model, hyperparameters_RF,
                               n_iter=50, cv=5, scoring='r2',
                               random_state=0)
rf_search.fit(X_train, y_train)
print(rf_search.best_params_)
```

Script 6.8 Recherche aléatoire des hyperparamètres de RF

Cette recherche retourne les hyperparamètres suivants : *n_estimators* = 100, *max_depth* = 11 et *max_features* = 4. Il convient de noter que cette méthodologie de travail (régularisation et optimisation avec *RandomizedSearchCV*) est à privilégier à des recherches exhaustives type *GridSearchCV*. En effet, la régularisation permet de trouver un intervalle optimale des hyperparamètres ce qui nous permet d'éviter de lancer des recherches exhaustives avec une consommation très importante du temps de calcul.

Les résultats du modèle optimisé (Figure 6.15c) montrent une très légère différence avec le modèle initial avec un RMSE qui passe de 0.67 à 0.65 et un R^2 constant de 0.92. Néanmoins, on observe que le modèle régularisé arrive également à des R^2 supérieurs à 0.9 pour l'ensemble de validation, éliminant ainsi toute crainte d'aléas statistique sur le lot de test. Donc, malgré le fait que le R^2 d'apprentissage est presque de 1, les résultats montrent bien un modèle capable de généraliser sur des nouvelles données avec une très bonne performance.

Optimisation de XGBoost

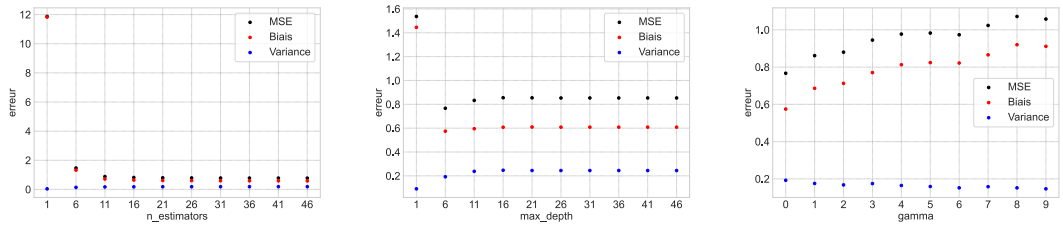
Pour régulariser l'algorithme XGBoost, on choisit d'optimiser les hyperparamètres suivants :

– *n_estimators* : nombre d'estimateurs (arbres) dans l'ensemble. La valeur par défaut est 100. D'une manière générale, un nombre plus grand d'estimateurs limite le sur-apprentissage mais augmente le temps de calcul.

– *max_depth* : profondeur maximale des estimateurs. La valeur par défaut est 6.

– *gamma* : paramètre de régularisation. Des valeurs grandes réduisent le risque de sur-apprentissage. La valeur par défaut est 0. Les valeurs autour de 20 sont très élevées (Laurae, 2016).

Selon les résultats observés dans la Figure 6.11, on trouve que les plages de valeurs optimales des hyperparamètres de XGBoost sont : *n_estimators* supérieur à 11, *max_depth* entre 6 et 11 et *gamma* acceptable jusqu'à environ 2.



(a) n_estimators (b) max_depth (c) gamma

Figure 6.11. Biais et variance de XGBoost en variant *n_estimators*, *max_depth* et *gamma*

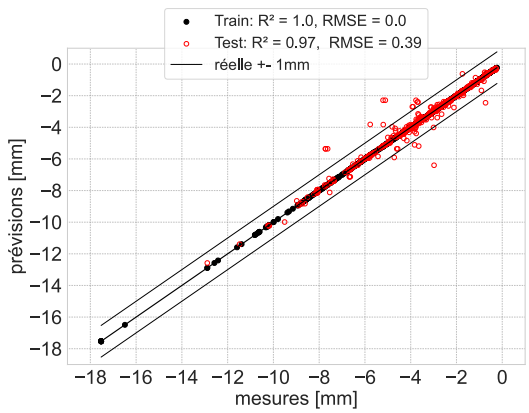
Par la suite, on effectue une recherche aléatoire dans la plage des hyperparamètres retrouvés en utilisant la fonction *RandomizedSearchCV*.

```
# {python}
hyperparameters_XGB = {'n_estimators': [11, 25, 50, 75, 100, 150],
                       'max_depth': range(6,12),
                       'gamma': [1, 2]}
model = XGBRegressor(seed = 0)
xgb_search = RandomizedSearchCV(model, hyperparameters_XGB,
                                n_iter=50, cv=5, scoring='r2',
                                random_state=0)
xgb_search.fit(X_train, y_train)
print(xgb_search.best_params_)
```

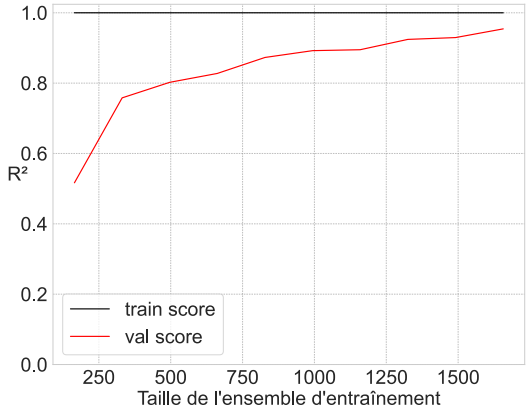
Script 6.9 Recherche aléatoire des hyperparamètres de XGBoost

Cette recherche retourne les hyperparamètres suivants : *n_estimators* = 50, *max_depth* = 6 et *gamma* = 1.

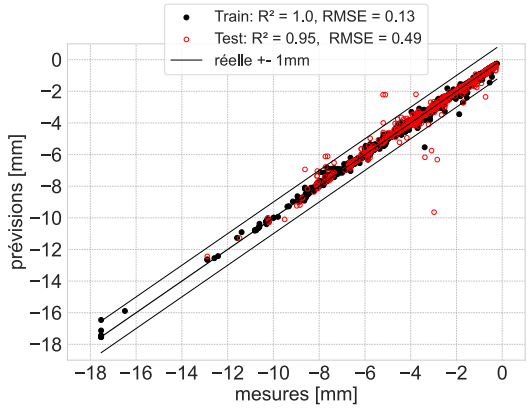
Les résultats du modèle optimisé (Figure 6.15e) montrent une très légère différence avec le modèle initial avec un R^2 qui passe de 0.93 à 0.9. Néanmoins, tout comme le modèle de RF, le modèle de XGBoost retourne des résultats satisfaisants (de l'ordre de 0.9 pour le R^2) sur l'ensemble de test ainsi que sur l'ensemble de validation. On peut donc conclure que le modèle est capable de généraliser même si le R^2 sur l'ensemble d'apprentissage effleure les valeurs de 1.



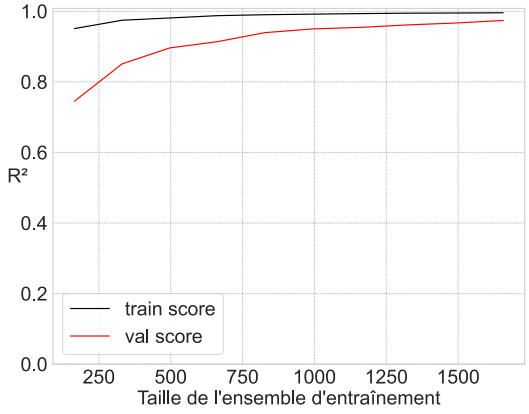
(a) DT



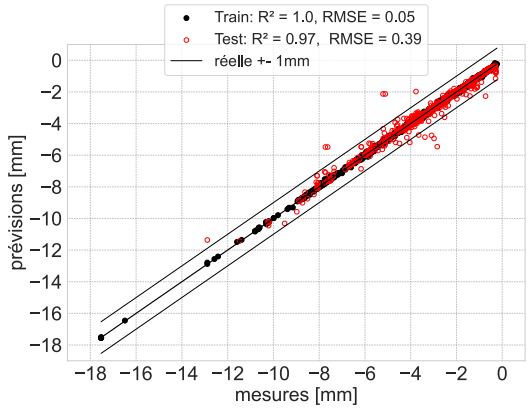
(b) Learning curves DT



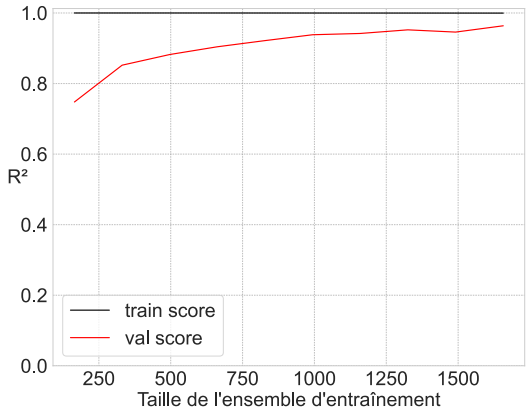
(c) RF



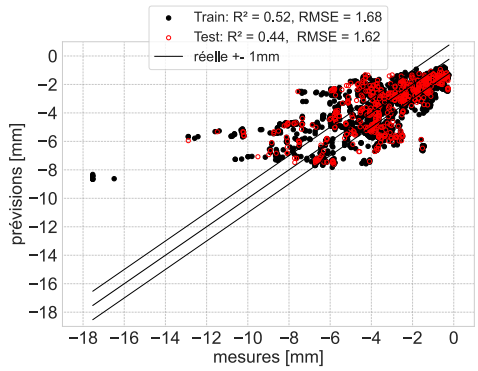
(d) Learning curves RF



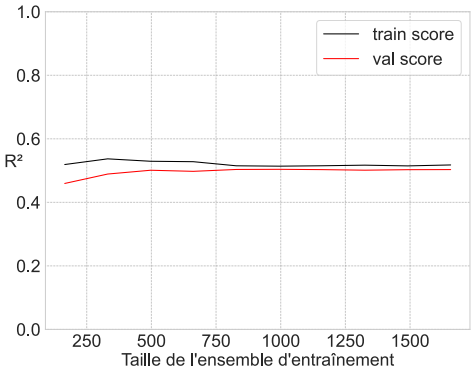
(e) XGBoost



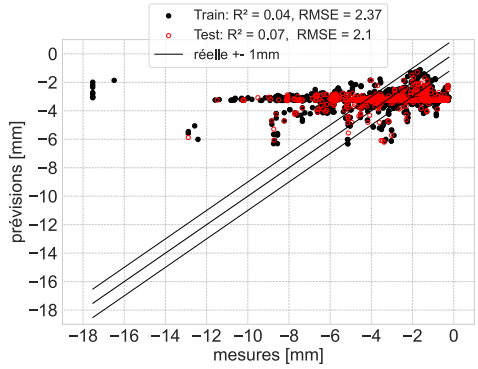
(f) Learning curves XGBoost



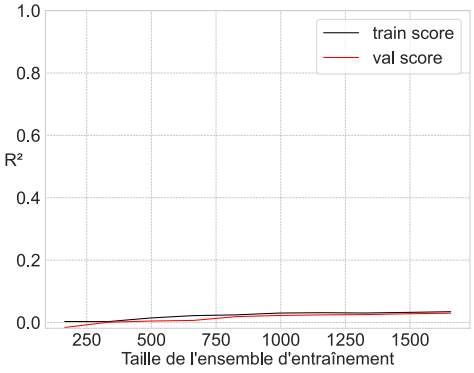
(g) LR



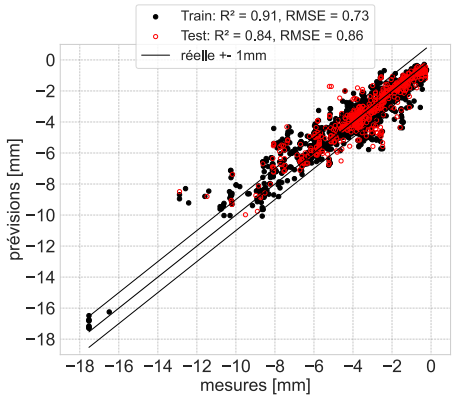
(h) Learning curves LR



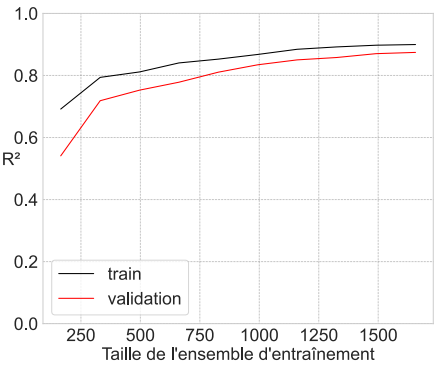
(i) SVM



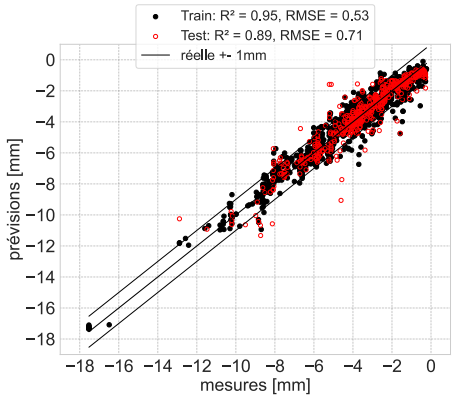
(j) Learning curves SVM



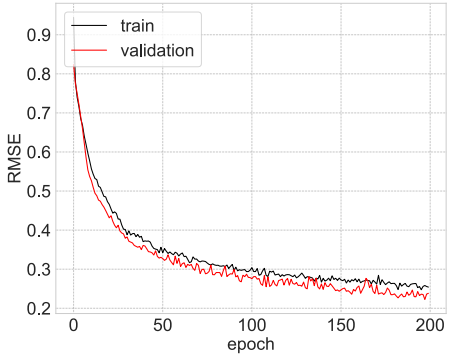
(k) SVM après standardisation



(l) Learning curves SVM après standardisation

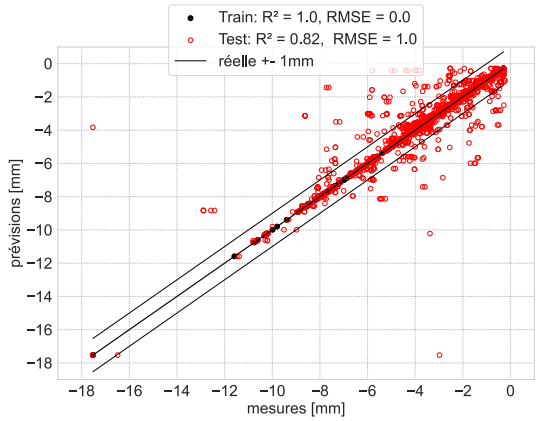


(m) BPNN

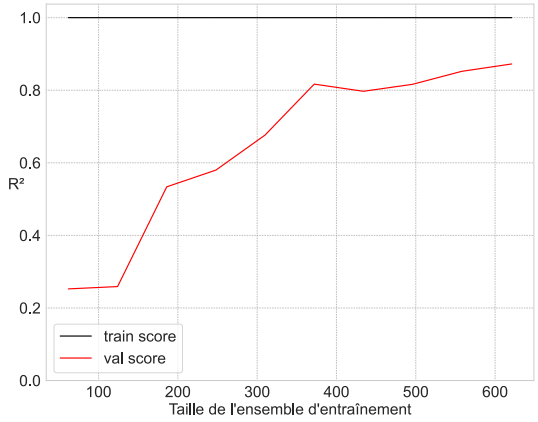


(n) Learning curves BPNN

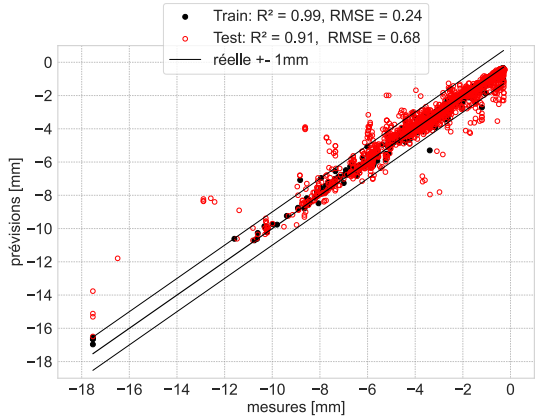
Figure 6.12. Résultats des modèles non optimisés (entraînement sur 80% des données)



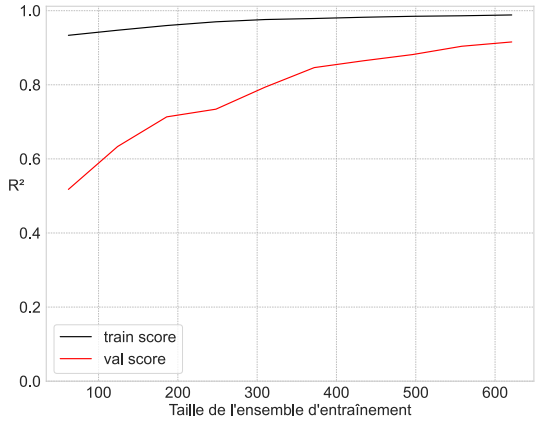
(a) DT



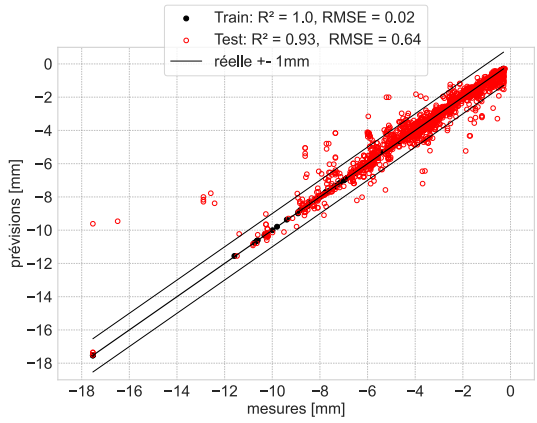
(b) Learning curves DT



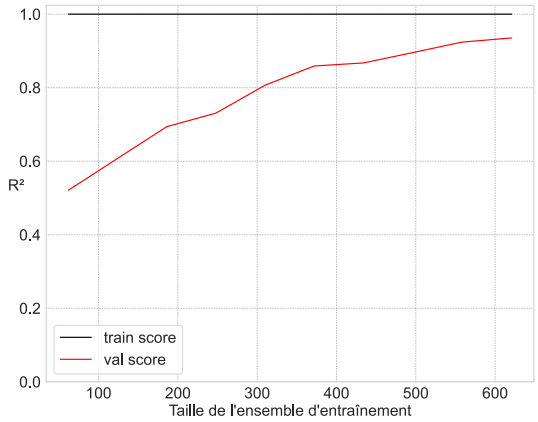
(c) RF



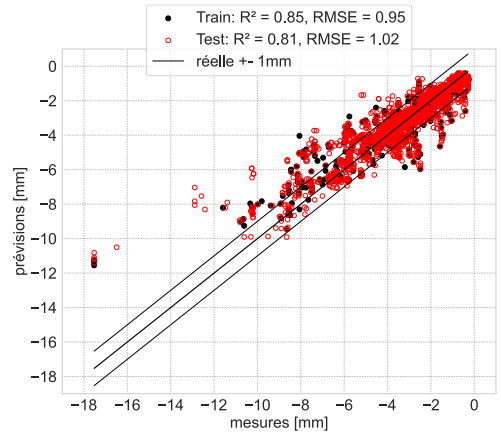
(d) Learning curves RF



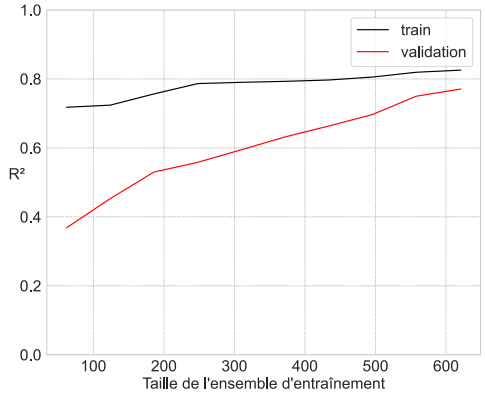
(e) XGBoost



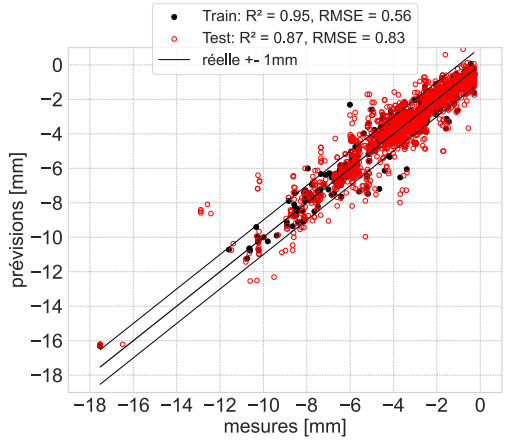
(f) Learning curves XGBoost



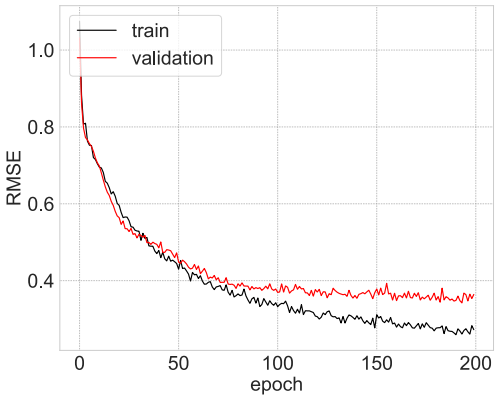
(g) SVM



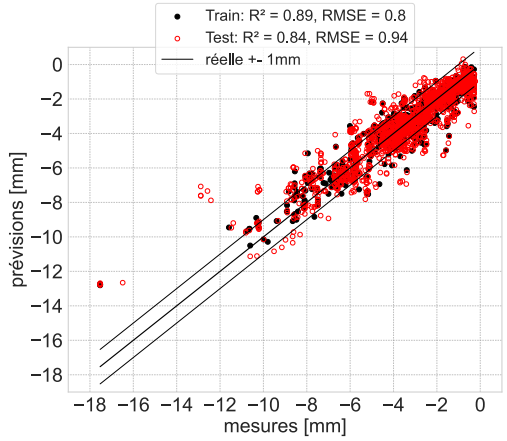
(h) Learning curves SVM



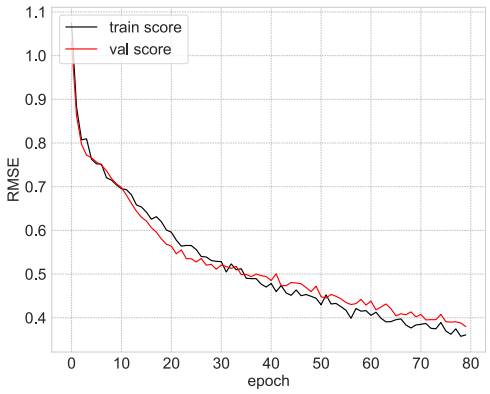
(i) BPNN



(j) Learning curves BPNN

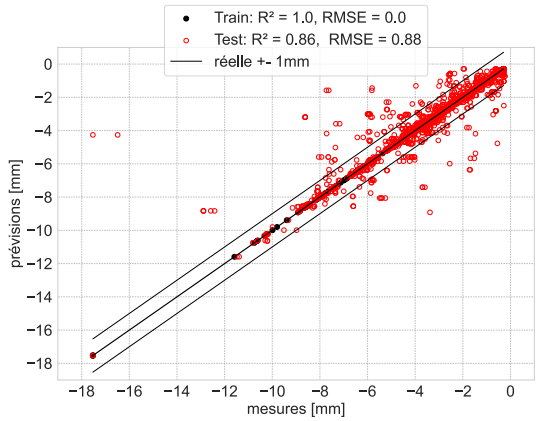


(k) BPNN (120 époques)

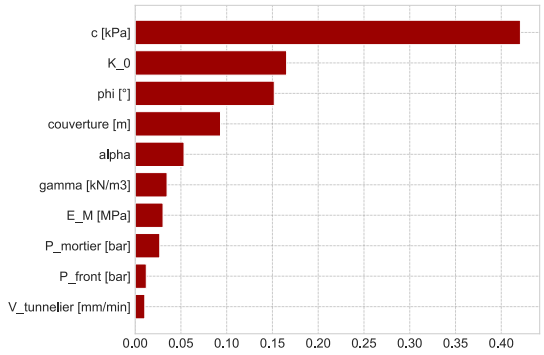


(l) Learning curves BPNN (120 époques)

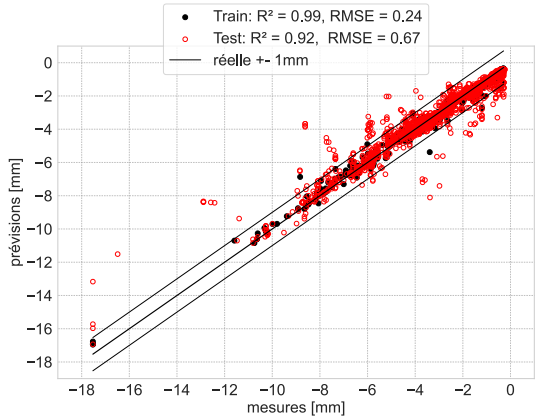
Figure 6.13. Résultats des modèles non optimisés (entraînement sur 30% des données)



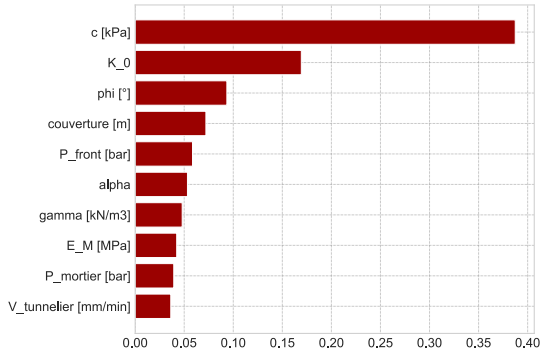
(a) DT



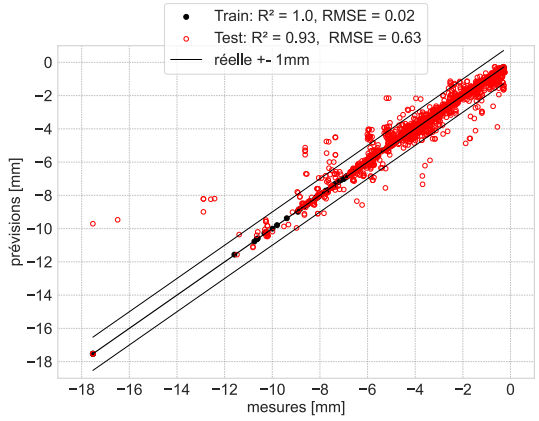
(b) Importance des caractéristiques, DT



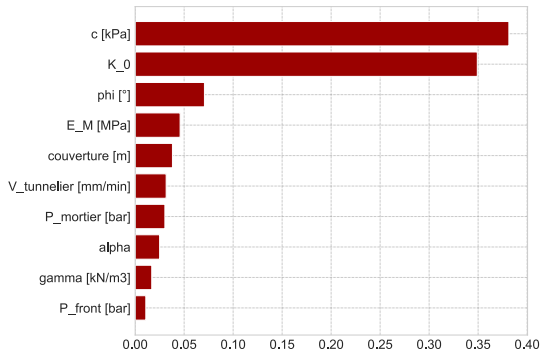
(c) RF



(d) Importance des caractéristiques, RF

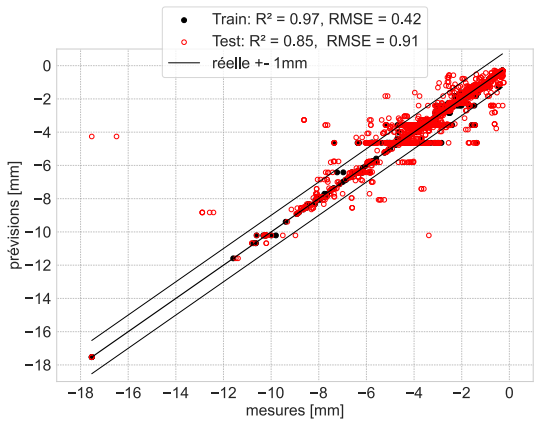


(e) XGBoost

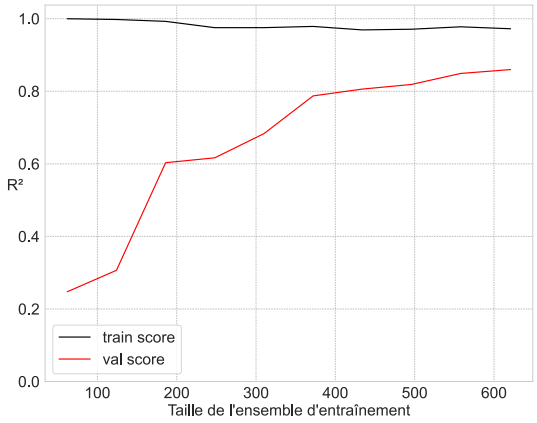


(f) Importance des caractéristiques, XGBoost

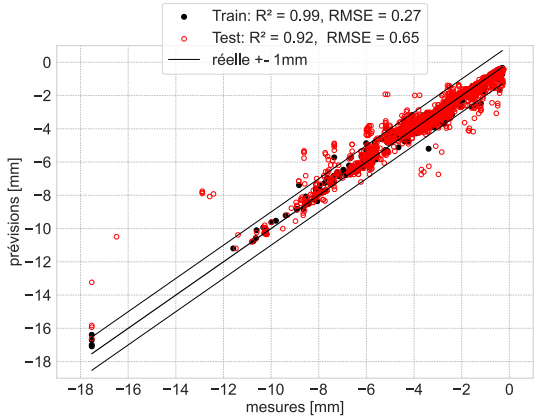
Figure 6.14. Résultats de DT, RF et XGBoost après l'optimisation des caractéristiques (entraînement avec 30% des données)



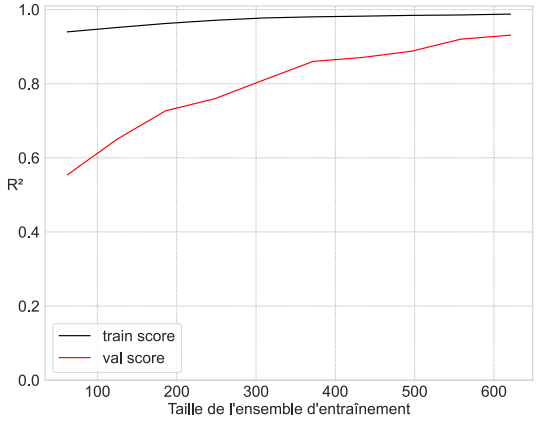
(a) DT



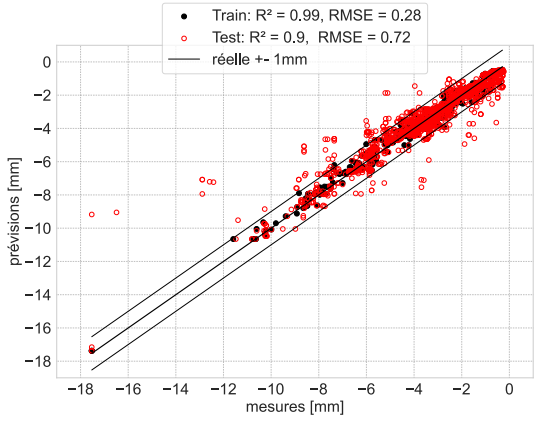
(b) Learning curves DT



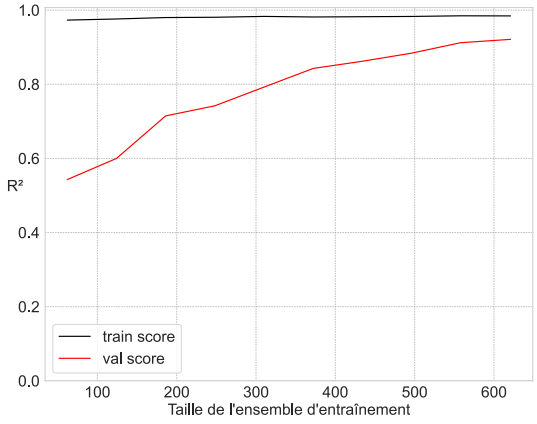
(c) RF



(d) Learning curves RF



(e) XGBoost



(f) Learning curves XGBoost

Figure 6.15. Résultats des algorithmes DT, RF et XGBoost après régularisation des hyperparamètres

6.3 Prévision du tassement maximal à une distance de l'axe

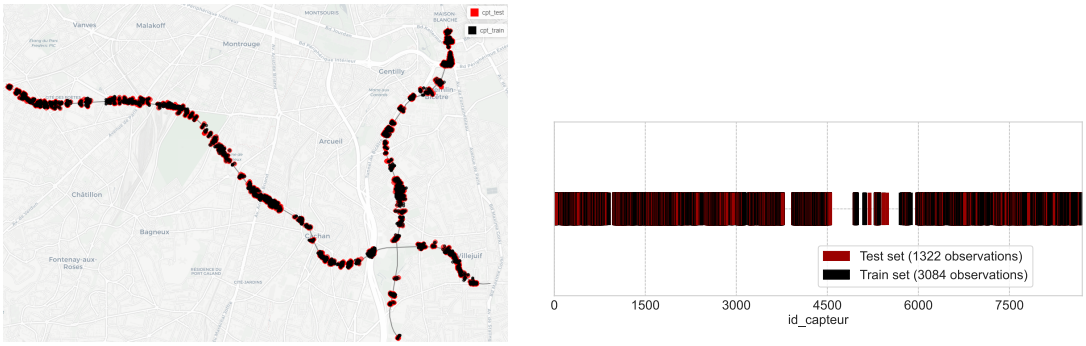
Nous présentons dans cette partie l'exercice de prévision du tassement maximal à une distance donnée de l'axe du tunnel s^*. Cette approche est intéressante notamment parce quelle offre la possibilité de retrouver le tassement transversal sans passer par une estimation des paramètres géométriques i_y et m_y.

Nous présentons dans ce qui suit les résultats des modèles pour une division aléatoire des données ainsi que les travaux de régularisation et d'optimisation de l'algorithme le plus performant.

6.3.1 Résultats des modèles

L'ensemble de données contient 4 406 observations qui seront divisées aléatoirement en ensemble d'apprentissage et de test. Les proportions choisies sont 30% des observations sont utilisés pour le test et 70% pour l'apprentissage et la validation. Ce choix est fait en fonction de la taille de l'ensemble des données : 70% représente 3084 observations et 30% représente 1322 observations. Ces chiffres nous semblent corrects pour bien entraîner l'algorithme et évaluer le modèle obtenu.

La répartition des deux ensembles selon l'***id_anneau*** ainsi que sur les tracés des deux lignes est présentée dans la Figure 6.16. La distribution des paramètres dans l'ensemble d'apprentissage et de test est montrée dans la Figure 6.17.



(a) (b)

Figure 6.16. Répartition des données sur le tracé du tunnel pour une division de 30% de l'ensemble des données en test

Les résultats des différents algorithmes sont regroupés dans la Figure 6.21. On remarque que les modèles obtenus par LR, SVM (après standardisation des données) et DT ont des résultats médiocres avec des R^2 respectifs de 0.41, 0.67 et 0.64.

Concernant les réseaux de neurones, les résultats présentés montrent une bonne performance du modèle obtenu, soit un R^2 de 0.8. Néanmoins, les learning curves des réseaux de neurones montrent que le modèle est en sur-apprentissage puisque la courbe de validation dépasse la courbe d'apprentissage avec l'augmentation du nombre de cycles (epoch).

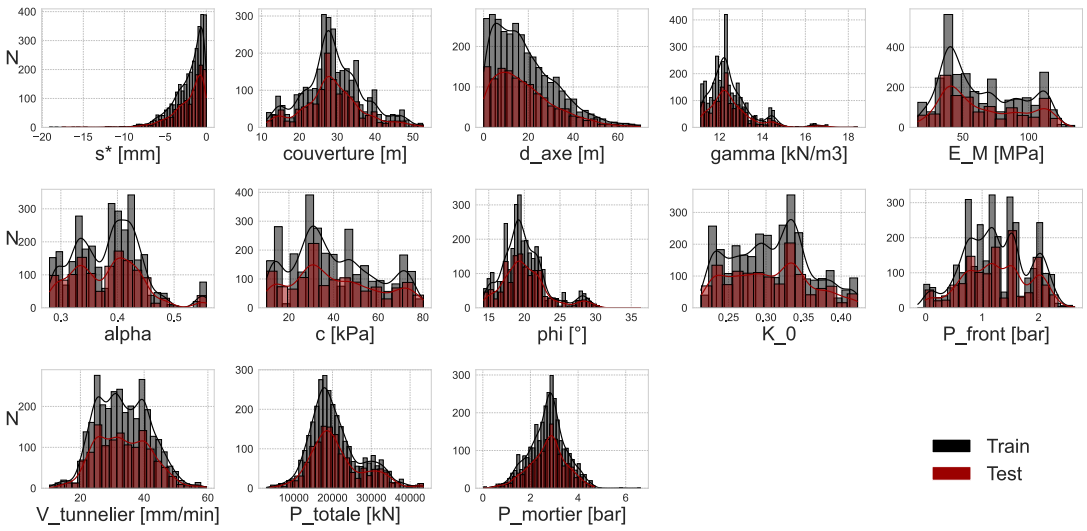
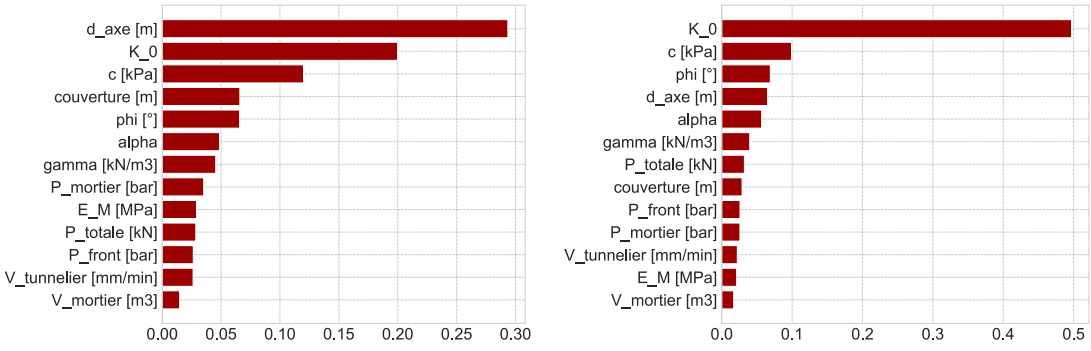


Figure 6.17. Distribution des caractéristiques avec une division de 20% de l'ensemble des données pour le test

Les modèles ensemblistes (RF et XGBoost) ont des R^2 de test de 0.83 et 0.82, respectivement. Cependant, ces modèles ont clairement fait du sur-apprentissage puisque les learning curves montrent des courbes d'apprentissage avec un R^2 de 1 pour un nombre très faible de données alors que la courbe de validation ne dépasse pas les 0.8 pour les deux algorithmes. Il faut donc procéder à la régularisation de ces modèles (RF et XGBoost), c'est-à-dire à l'optimisation des hyperparamètres pour éviter le sur-apprentissage.

6.3.2 Importance des caractéristiques

En suivant la même démarche que celle présentée précédemment, nous avons d'abord fait une optimisation du choix des caractéristiques. Pour cela, on regarde l'importance des caractéristiques pour les modèles obtenus par RF et XGBoost (Figure 6.18).



(a) RF (b) XGBoost

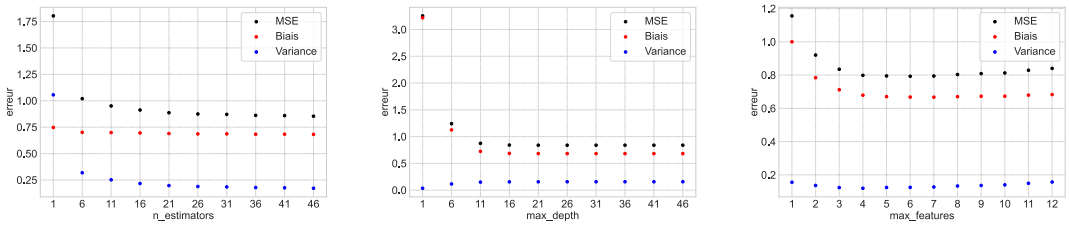
Figure 6.18. Importance des caractéristiques dans les modèle de RF et XGBoost

On trouve que le volume du mortier injecté à l'arrière de la jupe ($V_{mortier}$ [m^3]) a la plus faible influence dans les 2 modèles. Ce résultat n'est pas surprenant puisque les études statistiques, spécifiquement le calcul du coefficient de corrélation de Pearson R entre s^* et les paramètres d'entrées (Figure 5.31), montre que $V_{mortier}$ [m^3] a une corrélation très faible avec s^*. On décide alors d'éliminer ce paramètre dans la suite.

Il faut noter que la distance à l'axe du tunnel (d_{axe}) est la caractéristique avec la plus grande influence dans le modèle de RF, avec une valeur de presque 0.3, tandis que son importance ne dépasse pas la valeur de 0.1 dans le modèle de XGBoost (Figure 6.18). Ce dernier considère que K_0 (pour rappel, un paramètre combiné) est le seul paramètre ayant une vraie influence sur la valeur de s^*. En effet, toutes les autres caractéristiques ont une importance qui ne dépasse pas la valeur de 0.1. Ce comportement de XGBoost nous parait douteux, surtout que, physiquement, la distance d_{axe} est un paramètre ayant une grande influence sur la valeur de s^*. Pour en tirer des conclusions, il faudrait entraîner XGBoost avec un autre jeu de données pour vérifier que ce n'est pas simplement un coup de chance obtenu à cause de la technique du boosting (§ 2.3.1). Dans tous les cas, ces observations nous incitent à favoriser pour la suite le modèle de RF. Il est intéressant également de noter que RF donne plus de valeur aux paramètres combinés de sol, ce qui ne se voit pas si clair avec XGBoost.

6.3.3 Régularisation et Optimisation des hyperparamètres

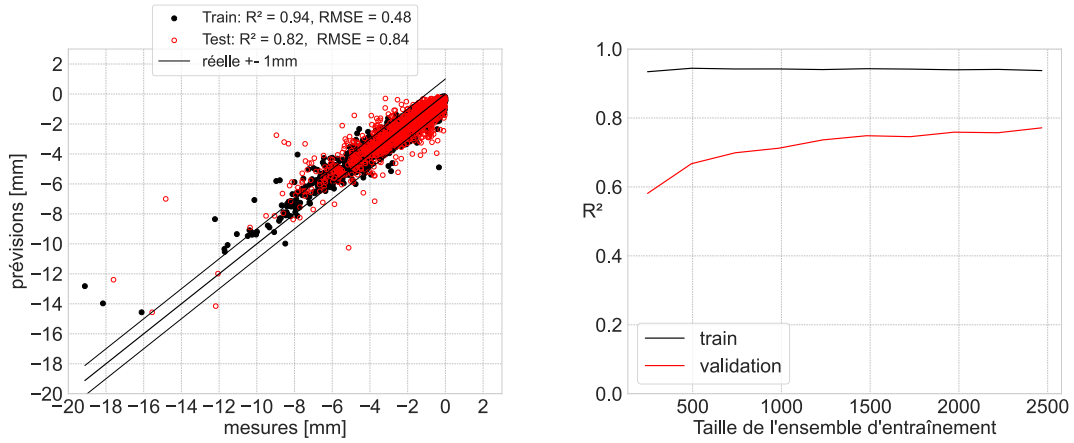
L'optimisation de RF suit la procédure décrite dans le § 6.2.3. Selon les résultats observés dans la Figure 6.19, on trouve que les plages de valeurs optimales des hyperparamètres de RF sont : *n_estimators* supérieur à 16, *max_depth* entre 6 et 11 et *max_features* entre 3 et 7.



(a) n_estimators (b) max_depth (c) max_features

Figure 6.19. Biais et variance de RF en variant *n_estimators*, *max_depth* et *max_features*

Par la suite, on effectue une recherche aléatoire dans ces plages des hyperparamètres en utilisant la fonction *RandomizedSearchCV* de la librairie *scikit-learn*. Cette recherche retourne les hyperparamètres suivants : *n_estimators* = 50, *max_depth* = 11 et *max_features* = 4.



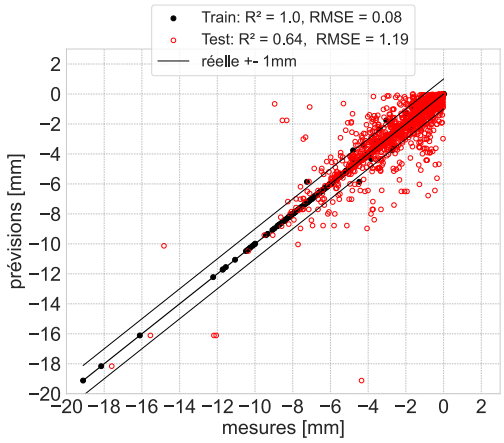
(a) RF

(b) Learning curves RF

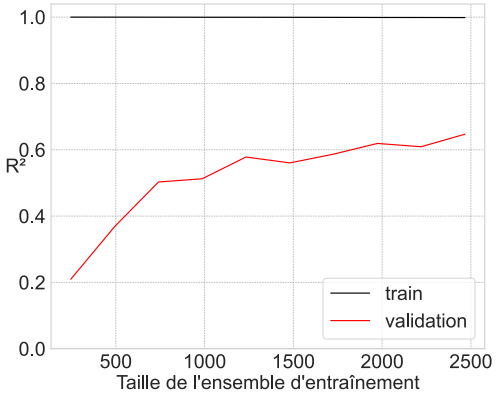
Figure 6.20. Résultats de l'algorithme RF après régularisation et optimisation des hyperparamètres

Les résultats du modèle optimisé (Figure 6.20) montrent une très légère différence avec le modèle initial avec un R^2 qui passe de 0.83 à 0.82 et un RMSE qui passe de 0.82 à 0.84 (détérioration). Malgré une performance légèrement réduite sur les données de test, les résultats restent satisfaisants après régularisation.

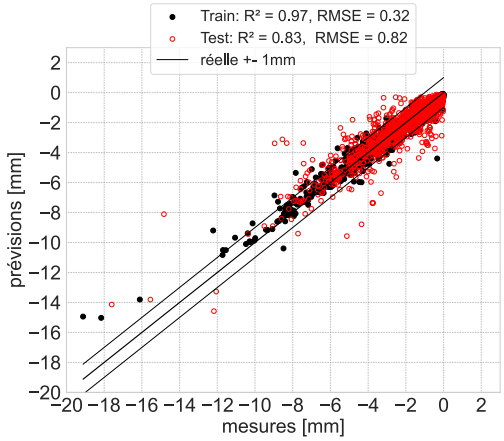
Il convient de noter que les résultats globaux obtenus lors de la prévision de s^* sont légèrement moins satisfaisants. Cela peut être expliqué par le fait que la prévision du tassement maximal en tout point est un problème plus complexe (rapport signal sur bruit plus faible).



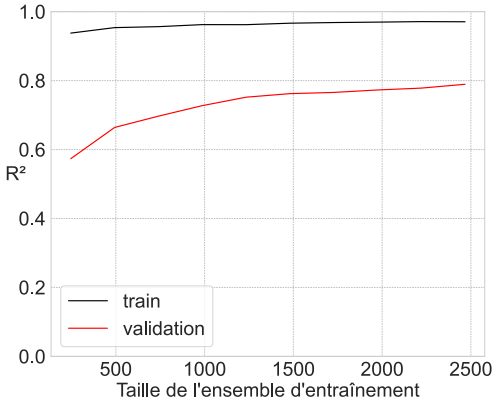
(a) DT



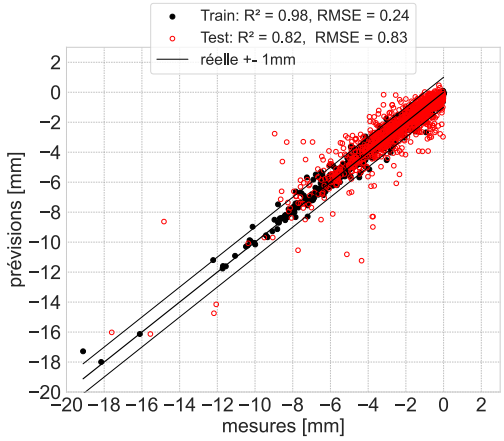
(b) Learning curves DT



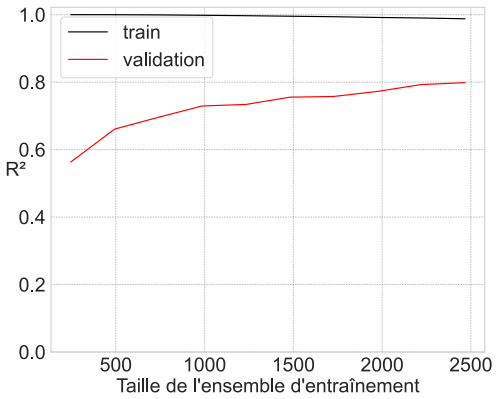
(c) RF



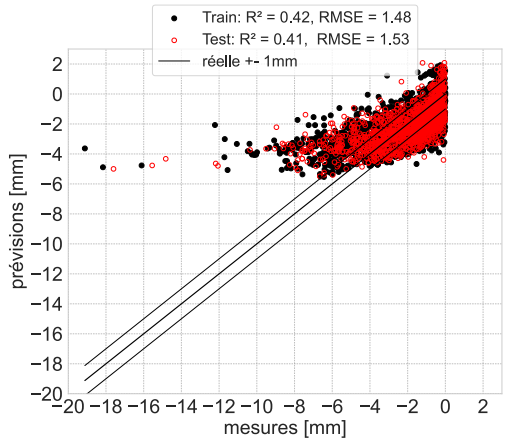
(d) Learning curves RF



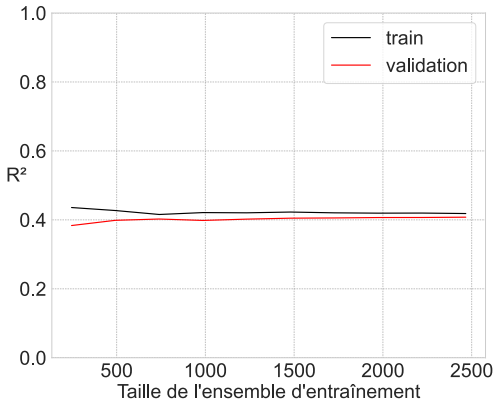
(e) XGBoost



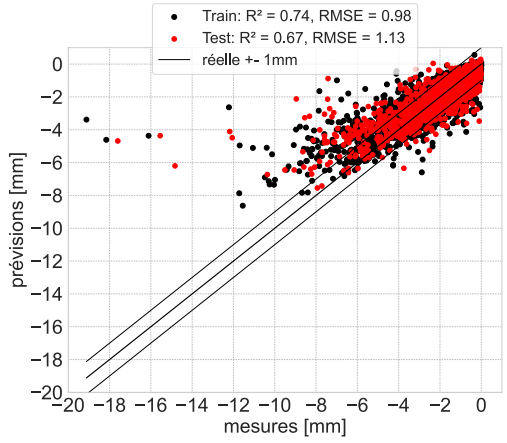
(f) Learning curves XGBoost



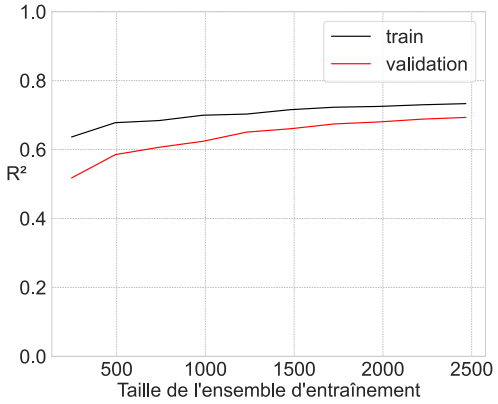
(g) LR



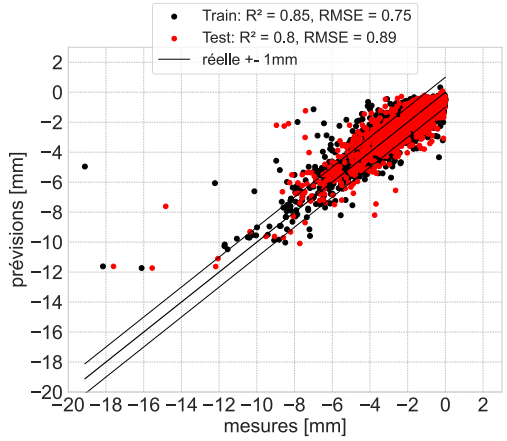
(h) Learning curves LR



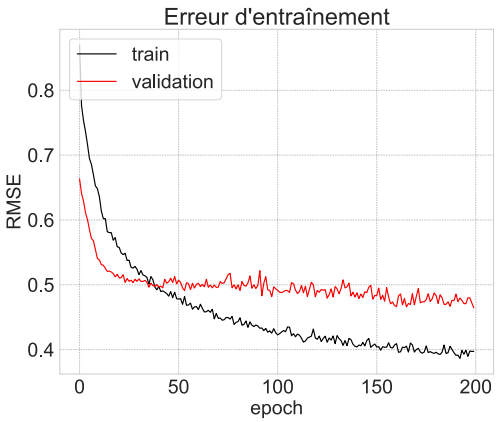
(i) SVM



(j) Learning curves SVM



(k) BPNN



(l) Learning curves BPNN

Figure 6.21. Résultats des modèles non optimisés (prévision de s^*)

Conclusion

Le présent chapitre était composé de trois parties. Tout d'abord, une explicitation de la méthodologie d'évaluation des modèles a été proposée.

Ensuite, on a présenté la prévision du tassement maximal à l'axe du tunnel (s_{max}) à l'aide de plusieurs algorithmes d'apprentissage automatique. Les algorithmes à base d'arbres de décision (RF et XGBoost) se sont avérés être les plus performants. Nous avons donc procédé à la régularisation et l'optimisation de leurs hyperparamètres afin d'obtenir des résultats plus fiables et maîtrisés, notamment vis-à-vis du sur-apprentissage.
Une autre conclusion importante de cette application est qu'on n'a pas besoin d'une grande quantité de données pour entraîner les algorithmes testés, notamment RF, XGBoost et BPNN. En effet, nous avons effectué des entraînements avec 777 observations de s_{max} et les résultats obtenus par ces modèles se sont avérés très satisfaisants. Ce chiffre reste assez élevé par rapport à ce qui est utilisé aujourd'hui dans l'état de l'art. Il convient de rappeler que Liu et al. (2022) sont parvenus à une conclusion similaire mais avec un ensemble de données de 187 observations.

Enfin, le même exercice est répété pour la prévision du tassement maximal à une distance de l'axe du tunnel (s^*). Cette approche est intéressante notamment pour la possibilité qu'elle offre de retrouver le tassement transversal sans passer par les paramètres géométriques i_y et m_y. Les résultats montrent que le modèle de RF est le plus fiable puisqu'il présente des résultats performants (en termes de R^2 et RMSE) et cohérents avec les connaissances métiers.

Les meilleures performances sont celles obtenues par RF lors de la prévision de s_{max}. Cet algorithme sera alors utilisé dans la partie suivante avec les hyperparamètres et les caractéristiques optimisés pour la prévision de s_{max}. L'objectif étant de tester les capacités de RF à prévoir le tassement s_{max} à l'avant du front.

Mise en œuvre sur des cas pratiques

7

Introduction

La prévision des tassements à venir au fur et à mesure du creusement au tunnelier ne peut se faire à partir d'un mélange aléatoire des données. En effet, une division aléatoire signifie qu'on prend des instances de test aléatoirement sur tout le tracé de la ligne. Or, lors d'un cas réel de creusement de tunnel, on ne s'intéresse pas à la prévision des tassements à l'arrière du front. On dispose a contrario des données collectées à l'arrière du front, et on veut prévoir ce qu'il va se passer lors de la suite du creusement. Pour cela, il est donc important de prendre en compte l'aspect spatio-temporel du problème lors du choix des données d'apprentissage et de test pour la prévision du tassement maximal à l'axe du tunnel (s_{max}). La capacité du modèle à prévoir sur ces zones non creusées sera nommée « extrapolation spatiale ». Il convient de différencier cette expression de l'extrapolation au sens mathématique, nommée ici « extrapolation statistique », qui est la prévision de la variable cible à partir de plages de données non connues dans l'ensemble d'apprentissage introduit à l'algorithme d'apprentissage automatique.

Dans un premier temps, on prend uniquement en compte l'aspect spatial. Pour ce faire, on choisit une distance de creusement à partir de laquelle les données sont séparées en un ensemble d'apprentissage et un ensemble de test. Toutes les données avant cette distance de creusement sont utilisées pour l'apprentissage, tandis que les données postérieures sont réservées à l'ensemble de test. Dans un second temps, on lance un entraînement en prenant en compte, en plus de la composante spatiale, la composante temporelle, liée au fait que les données sont un flux acquis progressivement dans le temps.

7.1 Prévisions sur une partie isolée d'un linéaire

Pour prendre en compte l'aspect progressif du creusement des tunnels, une division non aléatoire des données s'impose. L'objectif est d'obtenir un modèle qui apprend sur une partie creusée et prévoit le tassement sur une certaine partie de la suite du tracé.

Il convient de noter que l'algorithme utilisé dans cette partie est RF avec les hyperparamètres et les caractéristiques optimisés dans la partie précédente. On remarque également que le score R^2 est un mauvais indicateur en cas de faible quantité de données. En effet, l'équation de R^2 (Équation 2.2) est très sensible aux petits écarts de valeurs entre les données réelles et leur moyenne, et cela augmente avec la diminution de la taille des données. Il est donc préférable de se focaliser sur le score RMSE qui prend uniquement en

compte la différence entre les données réelles et les données prédites (Équation 2.6) ainsi que sur les visualisations.

7.1.1 Apprentissage sur le début d'un tronçon et prévision de sa fin

Il convient de rappeler que les profils en long des 3 tronçons étudiés dans cette partie (TR1, TR2 et L14S2) sont présentés dans la Figure 4.1 ainsi que dans la Figure 5.3 avec la distribution des paramètres mécaniques des sols combinés. On rappelle également que chaque ***id_anneau*** représente une position le long du tracé du tunnel et que la distance entre deux ***id_anneau*** consécutifs est d'1 m.

Apprentissage sur le début du TR1 et prévision de sa fin

Un premier test est lancé sur le tronçon TR1 de la L15SO. Ce tronçon s'étend entre les ***id_anneau*** 1 et 3938. On entraîne l'algorithme sur les ***id_anneau*** de 1 à 3000, puis on teste le modèle obtenu sur les ***id_anneau*** 3001 à 3938. Concrètement, cela représente un creusement établi sur 3000 m et une prévision des tassements sur le reste du tronçon, soit 938 m. Pour diviser les données, la fonction *train_test_split* n'est pas utilisée puisqu'on ne cherche plus à diviser les données selon un pourcentage, mais plutôt sur des longueurs.

```
# {python}
id_anneau_train = range(1, 3001)
id_anneau_test = range(3001, 3939)
# on n'a pas de valeurs de smax pour chaque id_anneau
id_anneau_train = [x for x in id_anneau_train if x in all_data.id_anneau
    .values]
id_anneau_test = [x for x in id_anneau_test if x in all_data.id_anneau.
    values]
X_train = all_data[all_data.id_anneau.isin(id_anneau_train)][features]
X_test = all_data[all_data.id_anneau.isin(id_anneau_test)][features]
y_train = all_data[all_data.id_anneau.isin(id_anneau_train)]["smax [mm]"
    ]
y_test = all_data[all_data.id_anneau.isin(id_anneau_test)]["smax [mm]"]
# melange des donnees d'entrainement
X_train, y_train = shuffle(X_train, y_train, random_state=0)
```

Script 7.1 Division des données sur des longueurs

Les résultats sont présentés dans la Figure 7.2. Tout d'abord, on note que l'apprentissage et le test sont effectués avec un nombre d'observation de 576 et 304, respectivement (Figure 7.2a). Il convient de rappeler qu'on n'a pas des valeurs de s_{max} sur tout le tracé puisque cela dépend de la distribution des capteurs en surface. Ensuite, le profil en long montre que la stratigraphie est plutôt homogène sur la zone d'apprentissage (Figure 7.2b). Le front est majoritairement mixte avec du Calcaire Grossier et des Argiles Plastiques. Cependant, à la fin de la zone d'apprentissage et dans la majorité de la zone de test, le

front du tunnel se trouve dans un mélange de Calcaires et Marnes de Meudon, d'Argiles Plastiques et de Craie. De plus, on voit l'apparition des carrières à ciel ouvert du Calcaire Grossier dans la zone de test. La visualisation suivante (Figure 7.2c) est la distribution de la variable cible, s_{max}, et des caractéristiques pour les deux ensembles d'apprentissage et de test. On remarque que les valeurs de s_{max} pour l'apprentissage ne dépassent pas 6 mm, alors que celles utilisées pour le test vont jusqu'à environ 12 mm. De même, la distribution des caractéristiques montrent des plages de valeurs très différentes entre les ensembles d'apprentissage et de test, notamment pour c [kPa], K_0, E_M [MPa] et α. Or, les résultats de la section précédente, spécifiquement la Figure 6.14, montrent que c [kPa], K_0 et φ [°] sont les caractéristiques qui ont la plus grande influence sur la prévision de s_{max} en utilisant l'algorithme RF.

Compte tenu de toutes ces différences entre l'ensemble d'apprentissage et l'ensemble de test, on en déduit qu'il faudrait que les modèles recherchés soient capables d'extrapoler sur des nouvelles plages de données (extrapolation statistique). Cependant, la discussion du 3.4.2 montre que les algorithmes d'apprentissage automatique sont conçus essentiellement pour l'interpolation et ne peuvent pas garantir des performances satisfaisantes en extrapolation. Donc, on peut déjà s'attendre à des résultats non satisfaisants avec ce degré d'extrapolation. En effet, la figure suivante (Figure 7.2d) montre que les scores de test sont respectivement -0.58 et 3.03 pour le R^2 et le RMSE. On rappelle que des valeurs négatives du coefficient de détermination R^2 indiquent une performance médiocre. Néanmoins, la courbe de validation montre un R^2 qui effleure la valeur de 1, ce qui signifie que le modèle est capable de généraliser (Figure 7.2e). Il convient de rappeler la différence entre généralisation et extrapolation : un modèle généralise s'il est capable de prévoir des valeurs dans des plages de valeurs déjà vues dans l'ensemble d'apprentissage, contrairement à l'extrapolation (statistique) qui cherche à prévoir des valeurs à l'extérieur de ces plages.

Pour aller plus loin, nous avons calculé une distance d'extrapolation spatiale raisonnable, qui correspond à la distance à partir du dernier ***id_anneau*** d'entraînement sur laquelle les prévisions sont acceptables. Cette distance à partir du dernier ***id_anneau*** est représentée par une échelle de couleur allant du rouge pour les premiers points de test (les plus proches de la zone d'entraînement) au violet pour les derniers points (cf. échelle de couleur à droite du graphique). On remarque sur la Figure 7.2d que pour les distances entre 0 et 300 m, soit les ***id_anneau*** entre 3000 et et 3300, les prévisions de s_{max} sont réussies avec une erreur d'environ 1 mm. Au-delà de cette distance, les valeurs s_{max} sont sous-évaluées. Ce phénomène peut s'expliquer par l'apparition des carrières à ciel ouvert de Calcaire Grossier à partir de l'***id_anneau*** 3300, ce qui a modifié considérablement les paramètres combinés de sol. Finalement, on conclut que le modèle est bien capable de généraliser sur des données dans la plage de valeurs connues mais incapable d'extrapolation statistique sur des nouvelles plages de valeurs.

Apprentissage sur 1000 m du TR1 et prévision sur les 2000 m suivants

L'application précédente consistait à entraîner un algorithme d'apprentissage automatique sur les données récupérées du creusement sur 3000 m et de prévoir les s_{max} sur les 938 m suivants. Nous avons conclu que le modèle est capable de généraliser sur les zones où les stratigraphies sont similaires à celles rencontrées dans la phase d'apprentissage. La problématique suivante est de trouver la distance de creusement suffisante pour entraîner un modèle performant ainsi que la distance jusqu'à laquelle la généralisation reste possible (distance d'extrapolation spatiale). La stratigraphie du tronçon TR1 étant similaire entre les ***id_anneau*** 1 et 3000 (pas d'extrapolation prévue), il est possible de lancer une multitude de tests pour trouver des réponses.

Le premier test consiste à prendre 1000 m de creusement en apprentissage et les 2000 m suivants en test. Nous avons donc entraîné l'algorithme avec les données entre l'***id_anneau*** 1 et 1000 pour ensuite tester le modèle obtenu sur les ***id_anneau*** 1001 à 3000. Les résultats sont présentés dans la Figure 7.3. Tout d'abord, on note que l'apprentissage et le test sont effectués avec un nombre d'observations de 173 et 403, respectivement (contre 576 et 304, précédemment) (Figure 7.3a). Ensuite, le profil en long montre que la stratigraphie est plutôt homogène sur la zone d'apprentissage et de test, la seule différence étant que le front du tunnel rencontre les Argiles Plastiques à la fin de la zone de test (Figure 7.3b). La visualisation suivante est la distribution de la variable cible, s_{max}, et des caractéristiques pour les deux ensembles d'apprentissage et de test (Figure 7.3c). On remarque que les valeurs sont bien distribuées entre les ensembles d'apprentissage et de test, sauf pour les paramètres de pilotage du tunnelier, notamment $V_{tunnelier}$ [mm/min] et P_{front} [bar]. La Figure 7.3d montre que les modèles ont très bien réussi à généraliser sur une distance d'environ 300 m, à partir de laquelle les valeurs de s_{max} commencent à être sur-estimées jusqu'à environ 1000 m. A partir de 1000 m, le modèle devient inadéquat et sous-estime les tassements de 2 à 4 mm. On conclut finalement que 1000 m de données de creusement pour l'apprentissage sont largement suffisants pour prévoir s_{max} correctement sur une distance d'environ 300 m.

Apprentissage sur 500 m du TR1 et prévision sur les 500 m suivants

L'exercice suivant consiste à diminuer encore plus la distance d'apprentissage et de tester sur 500 m. Nous avons donc choisi d'entraîner sur 500 m (***id_anneau*** entre 1 et 500) et tester le modèle sur les ***id_anneau*** entre 500 et 1000. Les résultats sont présentés dans la Figure 7.4. On note que l'apprentissage et le test sont effectués avec un nombre d'observations de 62 et 111, respectivement (contre 173 et 403 précédemment) (Figure 7.4a). Cette quantité de données semble très faible pour réussir à entraîner l'algorithme. Les résultats du modèle indiquent un R^2 de 0.17 et un RMSE de 0.76 (Figure 7.4d). Visuellement, on remarque que le modèle réussit à prévoir les valeurs de s_{max} à 1 mm près. Il faut noter que l'intervalle de tassement dans ces données est de $[\ -2.6\ mm\ ;\ -0.3\ mm]$,

donc 1 mm d'erreur représente entre 40% à 400% d'erreur, ce qui n'est pas négligeable. Par conséquent, ce modèle ne peut pas être considéré comme performant. Ceci est cohérent avec ce qui a déjà été vu puisque la distribution des paramètres (Figure 7.4c) montre des plages de valeurs différentes entre les ensembles d'apprentissage et de test, et que nous avons déjà établi que ces modèles sont incapables d'extrapoler statistiquement.

En conclusion, selon les études effectuées sur le tronçon TR1, on peut dire que l'algorithme des forêts aléatoires est très performant, capable de généraliser dans des plages de valeurs de paramètres similaires à celles rencontrées dans l'ensemble d'apprentissage même avec de faibles quantités de données (de l'ordre de la centaine). Cependant, cet algorithme semble incapable d'extrapoler statistiquement sur de nouvelles plages de données.

7.1.2 Apprentissage sur toute la L14S2 et 500 m d'un tronçon pour prédire la suite du tronçon

En pratique, l'expérience passée permet d'améliorer la conception lors de nouveaux projets. De façon similaire, il est intéressant ici de tester la performance de l'algorithme en l'entraînant sur les données d'une ligne déjà construite, avec des stratigraphies et une méthode de creusement similaires, puis d'ajuster l'entraînement sur le début d'un nouveau tronçon pour prévoir sa suite du creusement.

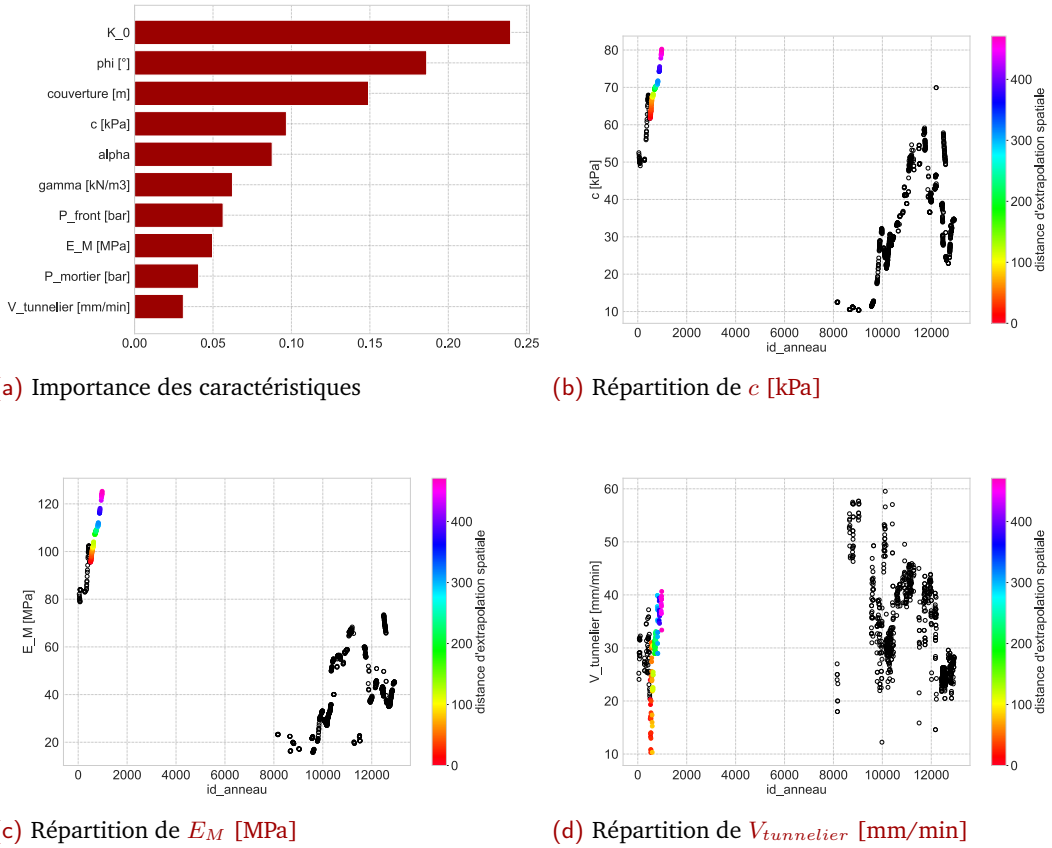
Apprentissage sur la L14S2 et 500m TR1 et prévision des 500m suivants du TR1

Dans l'exercice précédent, nous avons conclu qu'un apprentissage sur 500 m du TR1 pour prévoir les 500 m suivants ne donne pas de résultats satisfaisants. On se pose alors la question de l'effet de l'entraînement sur ces 500 m consécutivement à un entraînement sur les données de la L14S2. Dans cette étude, on entraîne l'algorithme sur toutes les mesures obtenues depuis la L14S2, soit 795 observations sur 5051 m, ainsi que sur les 500 premiers mètres du tronçon TR1 de la L15SO, soit 62 observations. Ensuite, on teste le modèle obtenu sur les 500 m suivants de TR1, soit 111 observations. Le profil en long de la L14S2 est présenté dans la Figure 4.1a.

Les résultats présentés dans la Figure 7.5 montrent que l'entraînement sur les données de la L14S2 améliore les résultats des points jusqu'à 300 m du début de la zone de test (les valeurs ne dépassent plus le seuil de 1 mm), mais détériore les résultats pour les points à partir de 350 m.

Pour mieux comprendre pourquoi certains points sont mal calculés, on vérifie si ce sont des points correspondant à une extrapolation statistique. On regarde alors la distribution des données d'entraînement et de test (Figure 7.5c). Il apparaît que ce sont majoritairement les caractéristiques c [kPa], E_M [MPa] et $V_{tunnelier}$ [mm/min] qui

ont des plages de valeurs dans l'ensemble de test qui n'existent pas dans l'ensemble d'apprentissage. On trace alors ces paramètres en fonction de la position et de la distance d'extrapolation spatiale (Figure 7.1). On remarque qu'en effet, les valeurs mal calculées



(a) Importance des caractéristiques

(b) Répartition de c [kPa]

(c) Répartition de E_M [MPa]

(d) Répartition de $V_{tunnelier}$ [mm/min]

Figure 7.1. Importance des caractéristiques et répartition de quelques caractéristiques en fonction de la position et de la distance d'extrapolation spatiale

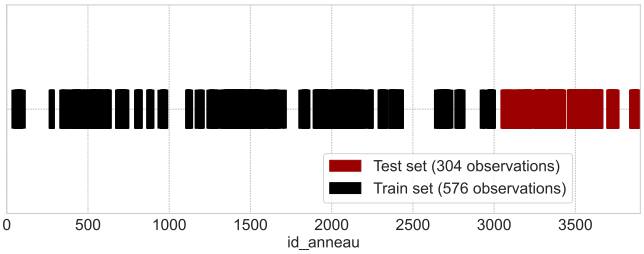
sont dans des régions qui n'ont pas d'équivalent dans la plage d'apprentissage pour c [kPa] et E_M [MPa] qui sont deux paramètres qui ont beaucoup d'influence sur les prévisions du modèle. En revanche, pour la caractéristique $V_{tunnelier}$ [mm/min], les points externes à la plage de mesures d'apprentissage sont les points en rouge qui ont été bien évalués par le modèle. Cela confirme, d'une part, que la caractéristique $V_{tunnelier}$ [mm/min] a une très faible influence sur les prévisions du modèles et, d'autre part, que l'algorithme utilisé est incapable d'extrapoler statistiquement.

7.1.3 Approches similaires sur un autre tronçon

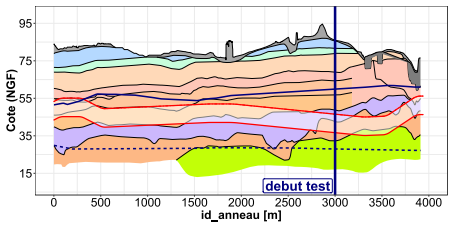
Pour éviter de conclure à partir du seul cas des données du TR1, un deuxième test est effectué cette fois-ci en prenant en compte les données du TR2. Il convient de noter que les 500 premiers mètres choisis pour le TR2 sont plus éloignés du début de la ligne pour éviter d'étudier des mesures de s_{max} affectés par les travaux de la gare d'Arcueil-Cachan de la L15SO ainsi que pour éviter la zone de la Bièvre qui est la zone avec la plus faible

profondeur du tunnel avec une importante variation des couches rencontrées au front du tunnel.

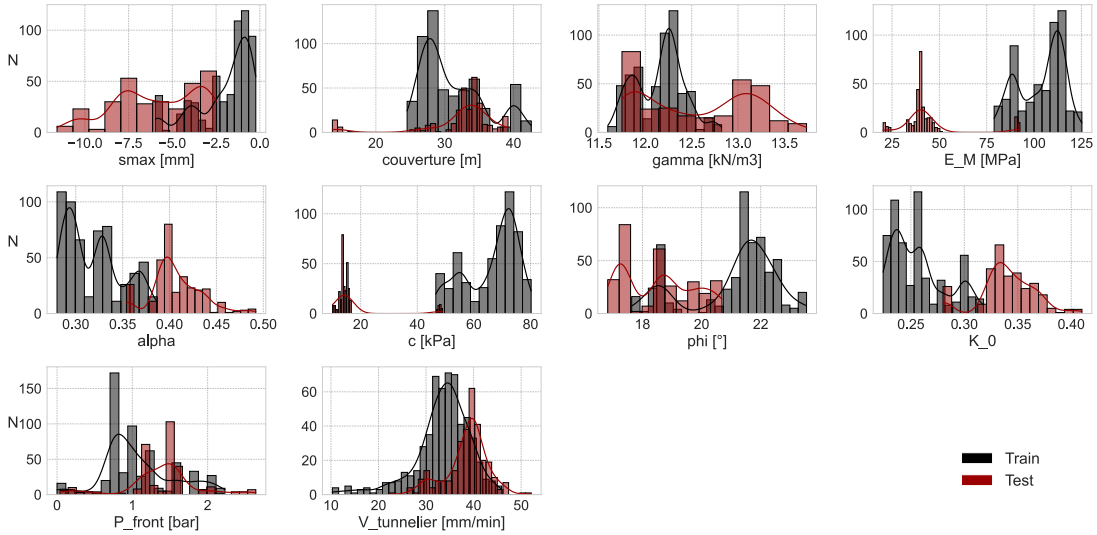
La Figure 7.6 et la Figure 7.7 présentent respectivement les résultats de l'entraînement sur 500 m du TR2 avec prévision du s_{max} sur les 500 m suivants et les résultats de l'entraînement sur la L14S2 en plus des 500 m du TR2 avec prévision sur les 500 m suivants du TR2.
Des conclusions similaires à celles obtenues avec les données du TR1 peuvent être tirées. D'une part, la quantité de données utilisée pour l'entraînement de RF sur uniquement 500 m n'est pas suffisante pour obtenir un modèle avec un taux d'erreur acceptable. En effet, on remarque que le taux d'erreur sur s_{max} est entre 25 et 100%. D'autre part, l'ajout des données de la L14S2 a nettement amélioré la prévision des s_{max} avec une distance d'extrapolation spatiale satisfaisante d'environ 350 m.



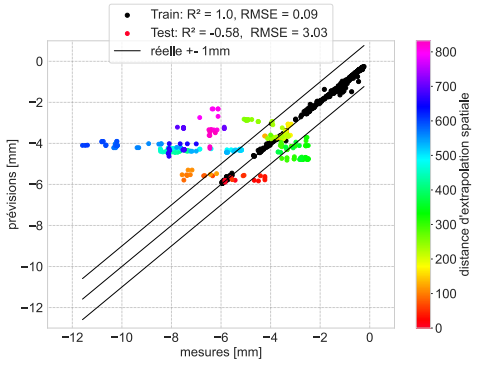
(a) Division des données



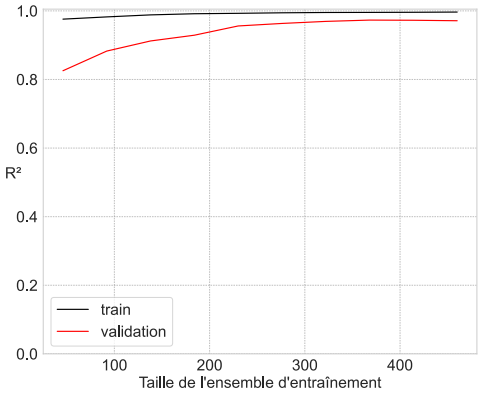
(b) Profil en long



(c) Distribution des données d'entraînement et de test

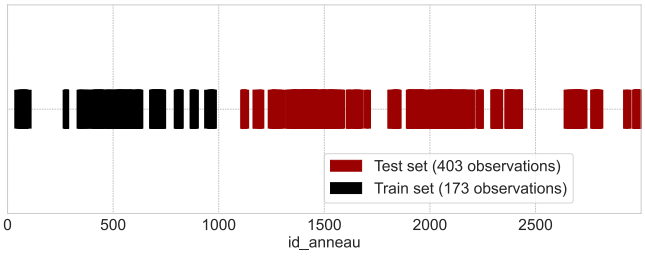


(d) RF

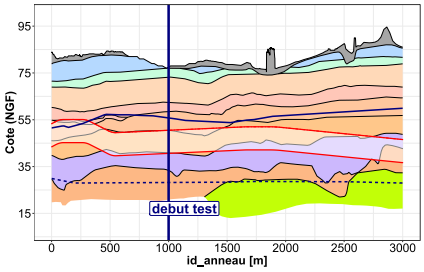


(e) Courbes d'apprentissages RF

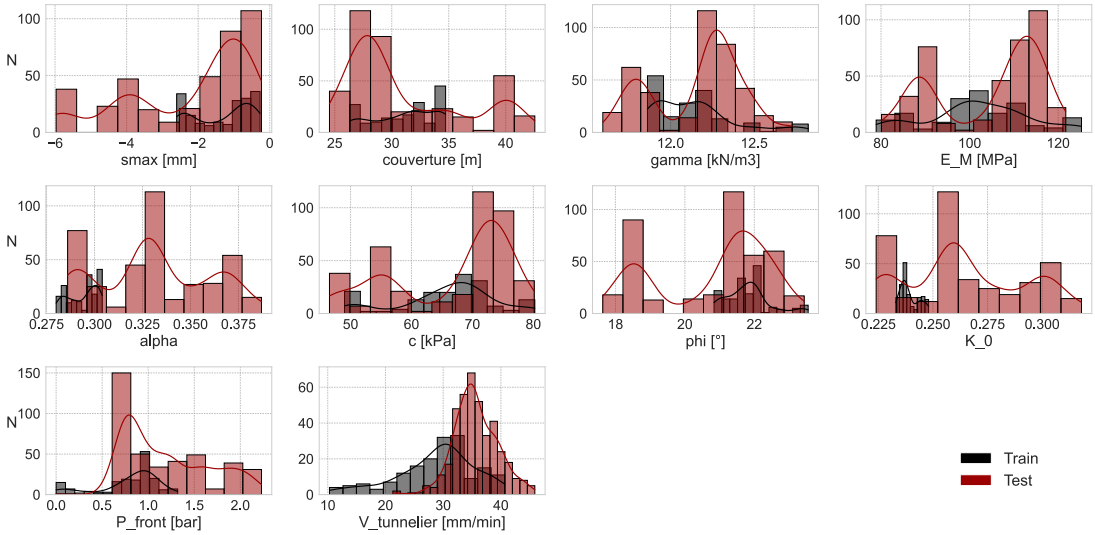
Figure 7.2. Résultats de la prévision des tassements en entraînant sur 3000 m et test sur les 938 m suivants du TR1



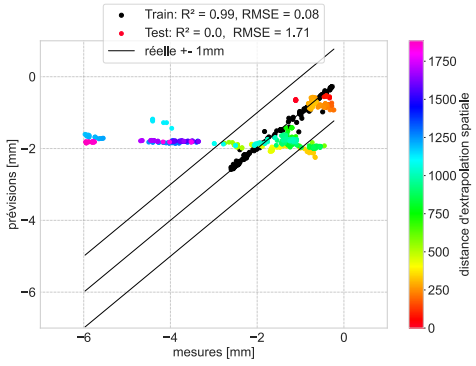
(a) Division des données



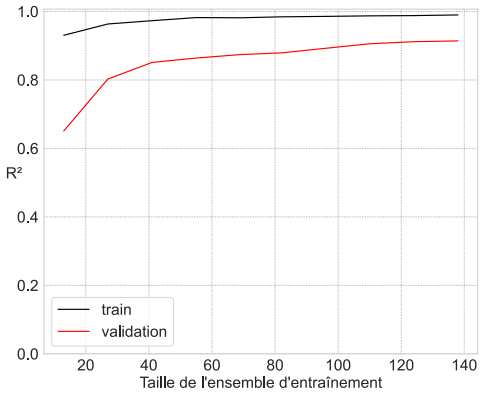
(b) Profil en long



(c) Distribution des données d'entraînement et de test

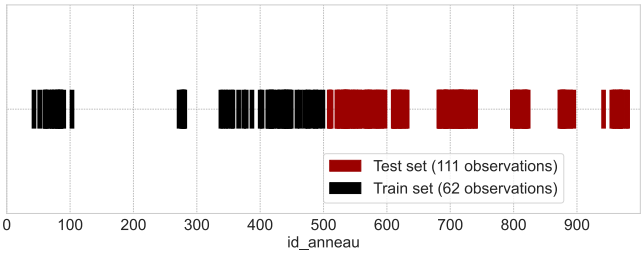


(d) RF

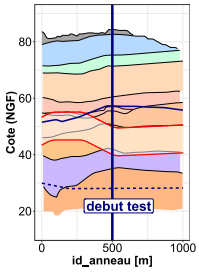


(e) Courbes d'apprentissages RF

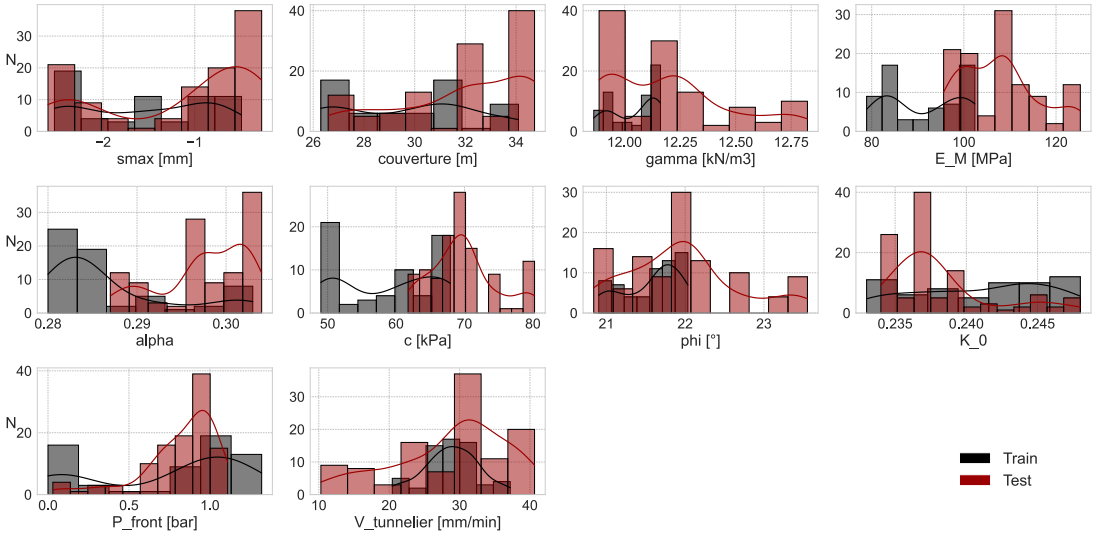
Figure 7.3. Résultats de la prévision des tassements en entraînant sur 1000 m et test sur les 2000 m suivants du TR1



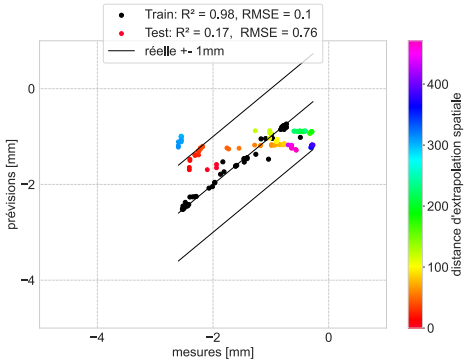
(a) Division des données



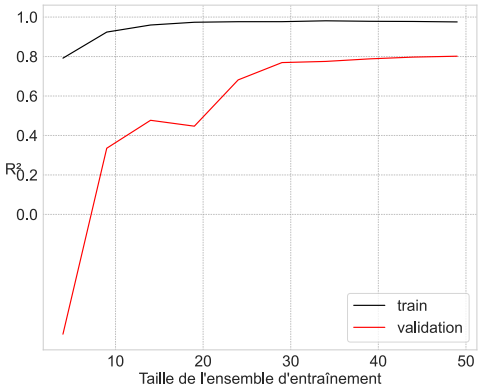
(b) Profil en long



(c) Distribution des données d'entraînement et de test

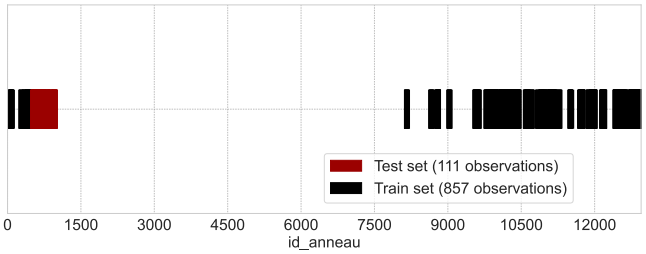


(d) RF

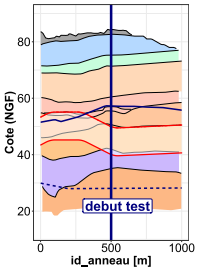


(e) Courbes d'apprentissages RF

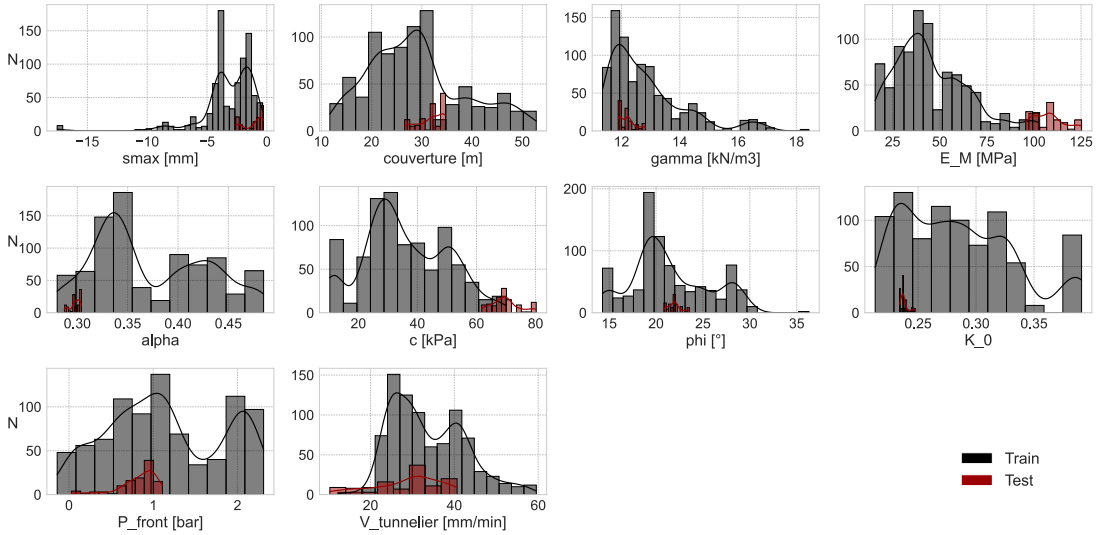
Figure 7.4. Résultats de la prévision des tassements en entraînant sur 500 m et test sur les 500 m suivants du TR1



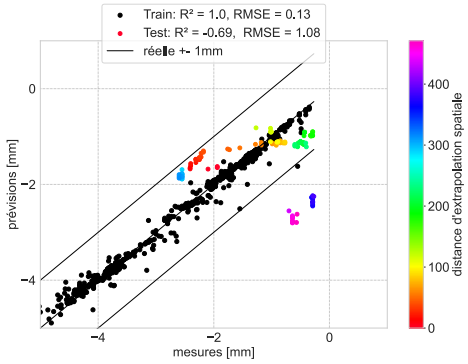
(a) Division des données



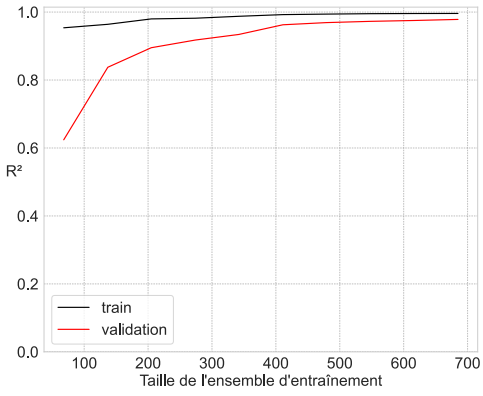
(b) Profil en long



(c) Distribution des données d'entraînement et de test

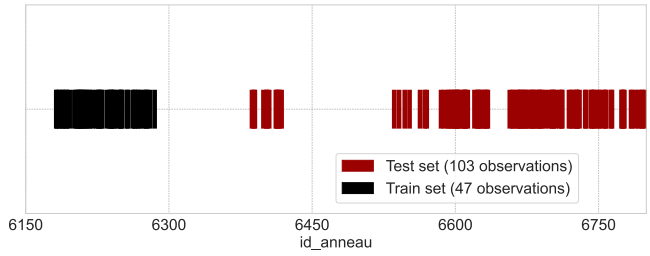


(d) RF

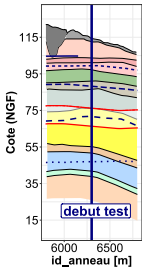


(e) Courbes d'apprentissages RF

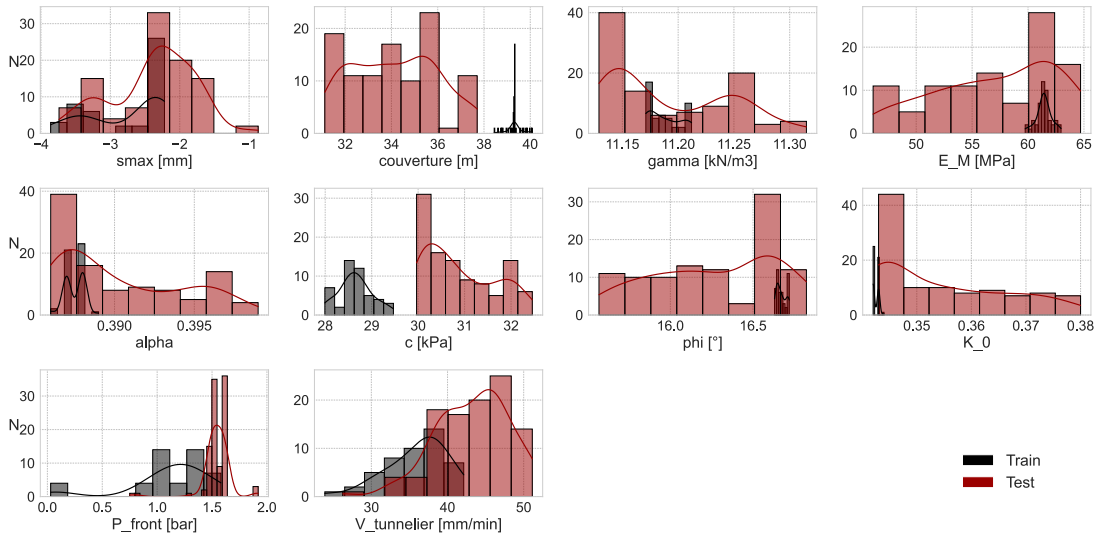
Figure 7.5. Résultats de la prévision des tassements en entraînant sur la L14S2 et 500 m du TR1 et test sur les 500 m suivants du TR1



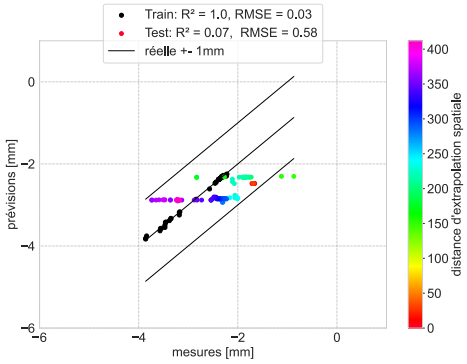
(a) Division des données



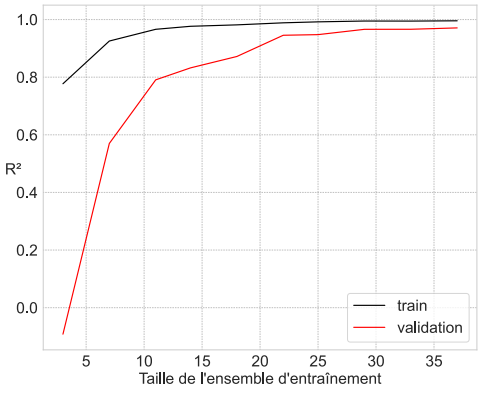
(b) Profil en long



(c) Distribution des données d'entraînement et de test

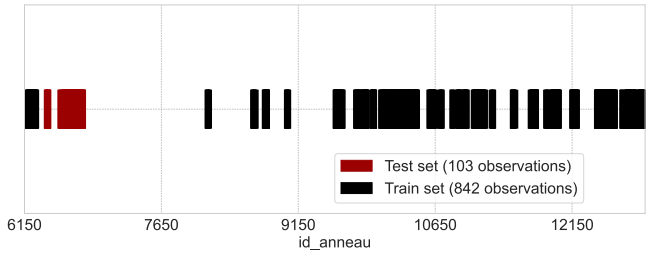


(d) RF

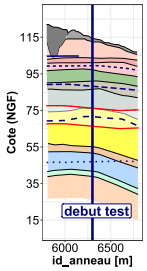


(e) Courbes d'apprentissages RF

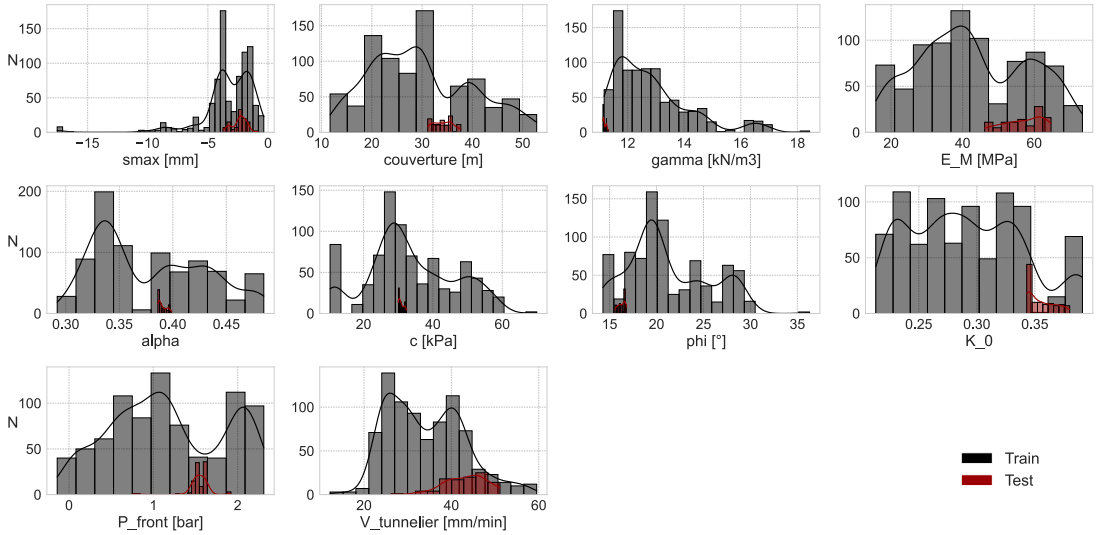
Figure 7.6. Résultats de la prévision des tassements en entraînant sur 500 m et test sur les 500 m suivants du TR2



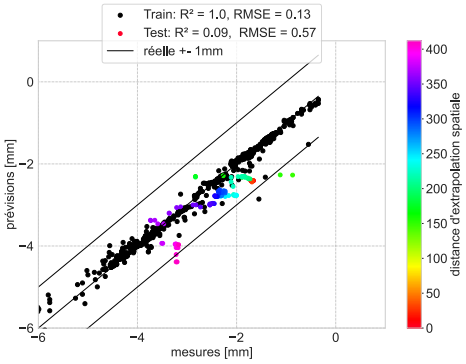
(a) Division des données



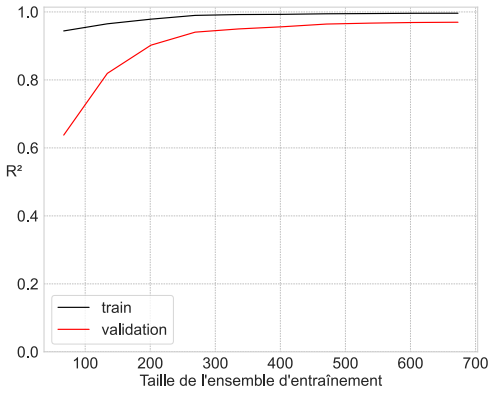
(b) Profil en long



(c) Distribution des données d'entraînement et de test



(d) RF



(e) Courbes d'apprentissages RF

Figure 7.7. Résultats de la prévision des tassements en entraînant sur la L14S2 et 500 m du TR2 et test sur les 500 m suivants du TR2

7.2 Mise en situation de la progression du creusement

Pour la suite, on cherche à développer un modèle simple capable de prédire le tassement maximal à l'axe du tunnel (s_{max}) en temps réel, i.e. pendant le creusement. En d'autres termes, on souhaite développer une approche capable de tenir compte de l'aspect spatio-temporel du problème. Par conséquent, on considère un mode de consommation des données par flux de données plutôt que par lot statique (§ 2.2.2), c'est-à-dire que l'entraînement est reproduit au fur et à mesure de la récupération de nouvelles données avec l'avancement du creusement. A savoir que l'algorithme d'apprentissage automatique utilisé est RF régularisé et optimisé, tel que présenté dans le § 6.2.3.

7.2.1 Modèle à partir d'un an de creusement

Pour commencer, prenons une date donnée, par exemple, 1 an après le début du creusement du premier tronçon, soit TR1. A cette date, les creusements suivants sont déjà effectués :

- Le tunnelier du TR1 a déjà creusé 2522 m et le front se trouve à l'***id_anneau*** 2522.
- Le tunnelier du TR2 a déjà creusé 1370 m et le front se trouve à l'***id_anneau*** 5309.
- Le tunnelier de la L14S2 a déjà creusé 1535 m et le front se trouve à l'***id_anneau*** 9670.

Nous voudrions prévoir les s_{max} à l'avant du front du tunnel du TR1, et donc à partir de l'***id_anneau*** 2522.

Pendant le creusement, nous n'avons évidemment pas accès aux valeurs s_{max} directement à l'arrière du front. Les tassements sont en train de se produire et vont mettre un certain temps à atteindre cette valeur. Cette zone de transition pourrait perturber les algorithmes d'apprentissage. L'idée est donc de neutraliser les tassements sur cette zone de transit des tassements de 0 à s_{max}. Selon les analyses statistiques proposées dans le § 5.3.1, on montre que, dans 75% des cas, 95% du tassement maximal est atteint au bout d'une distance de 75 m. Cette combinaison d'intervalles de confiance nous parait adaptée au choix d'une distance raisonnable de neutralisation des valeurs de tassements. Par précaution, nous avons retenu par la suite 100 m à l'arrière du front du tunnel. Nous avons alors décidé d'entraîner l'algorithme avec les s_{max} récupérés jusqu'à 100 m à l'arrière du front des trois tunnels (Figure 7.8). Cependant, les données de test doivent provenir d'un seul tronçon pour que des conclusions pertinentes vis-à-vis de la distance d'extrapolation affichée dans les résultats soient tirées. Nous avons choisi d'effectuer ce test sur le TR1 puisqu'il a une stratigraphie relativement homogène le long du profil. Cela permet de ne pas mettre, a priori, en difficulté les modèles d'apprentissage automatique, dont on a vu au chapitre précédent qu'ils sont incapables d'effectuer des extrapolations statistiques.

Les profils en long des zones de provenance des données d'entraînement et de test sont présentés dans la Figure 7.9. Les remarques suivantes peuvent être faites :

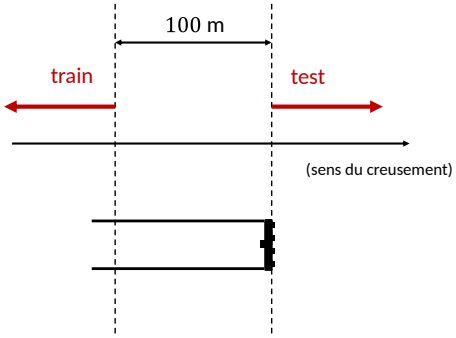
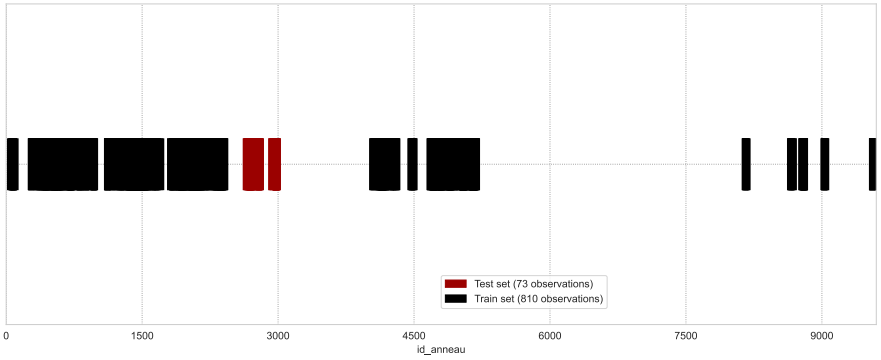


Figure 7.8. Division des données en ensembles d'apprentissage et de test en tenant compte de l'aspect spatio-temporel du problème

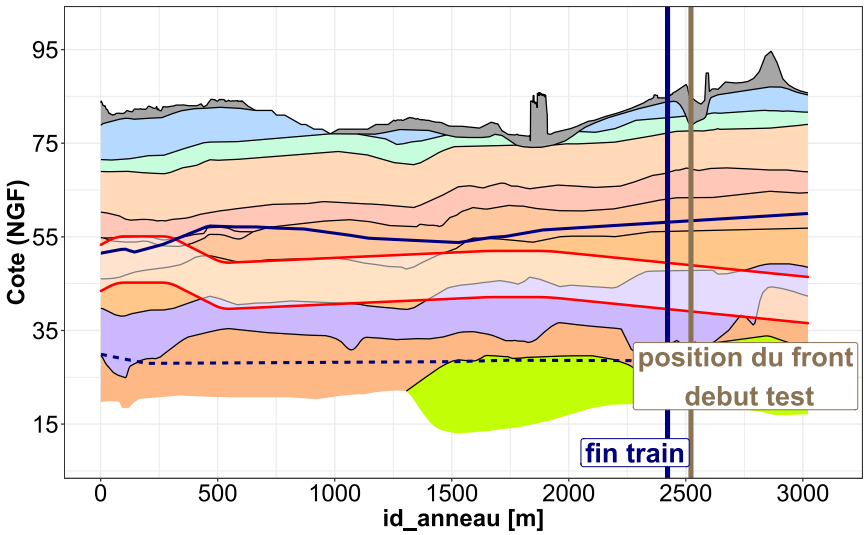
– Les sols rencontrés au front de la L14S2 sont très différents de ceux du TR1 et du TR2.
– Une similitude est détectée entre les couches rencontrées au front du TR1 et celles d'une partie du TR2 (mélange de Calcaire grossier avec Argile Plastique).
– Les sols rencontrés au front du tunnel du TR1 dans la zone de test sont ou bien de l'Argile Plastique, ou bien un mélange de Calcaires et Marnes de Meudon et d'Argiles Plastiques. Ce front mixte n'est pas rencontré dans les zones d'apprentissage ce qui pourrait induire de mauvaises prévisions dans cette zone.

Les résultats de l'entraînement dans ce cas d'étude sont présentés dans la Figure 7.10. Ces résultats indiquent que le modèle a réussi à effectuer une extrapolation spatiale jusqu'à une distance d'environ 400 m et cela avec une erreur inférieure à 1 mm, ce qui représente un taux d'erreur sur s_{max} entre 20 et 30%. Cela est cohérent avec les études effectuées dans la partie précédente (§ 7.1) qui montrent que la distance d'extrapolation spatiale est d'environ 300 m.

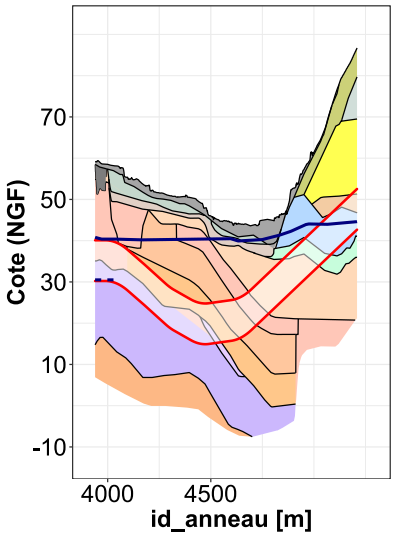
On peut approfondir l'analyse des résultats afin de déterminer l'origine des mauvaises prévisions au-delà de 400 m à l'avant du front. Tout d'abord, il convient de remarquer une transition brutale entre les mesures à une distance de 400 m qui sont très bien prévues et celles à une distance de 450 m qui ne sont pas correctement prévues. On peut avancer deux hypothèses couplées pour expliquer ce changement brutal : en premier lieu, il y a entre 400 et 450 m un défaut d'auscultation en surface (pas de capteurs posés dans la zone du Fort de Vanves). Ceci permet d'avancer qu'en réalité cette transition est bien progressive, mais que c'est l'absence de capteurs sur ce linéaire qui la fait apparaître brutale. En second lieu, ce changement coïncide avec deux modifications de stratigraphie : le passage autour de 300 m du front dans les Marnes de Meudon, et entre 300 et 400 m sous un remblai dans le fort de Vanves. Ces deux changements impliquent des dérives qui se compensent : les Marnes de Meudon n'ont jamais été vues par le modèle, elles sont raides, donc elles orientent le modèle vers la sous-estimation des tassements. Cependant, la présence des Remblais au début de l'apparition des Marnes compense cet effet, si bien que les tassements sont correctement prédits. Au moment où le front atteint une



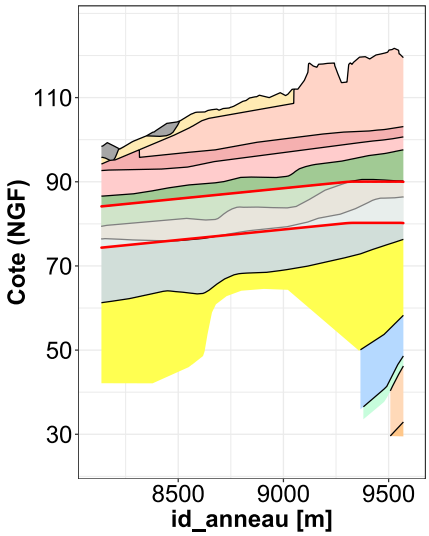
(a) Division des données



(b) Profil en long TR1

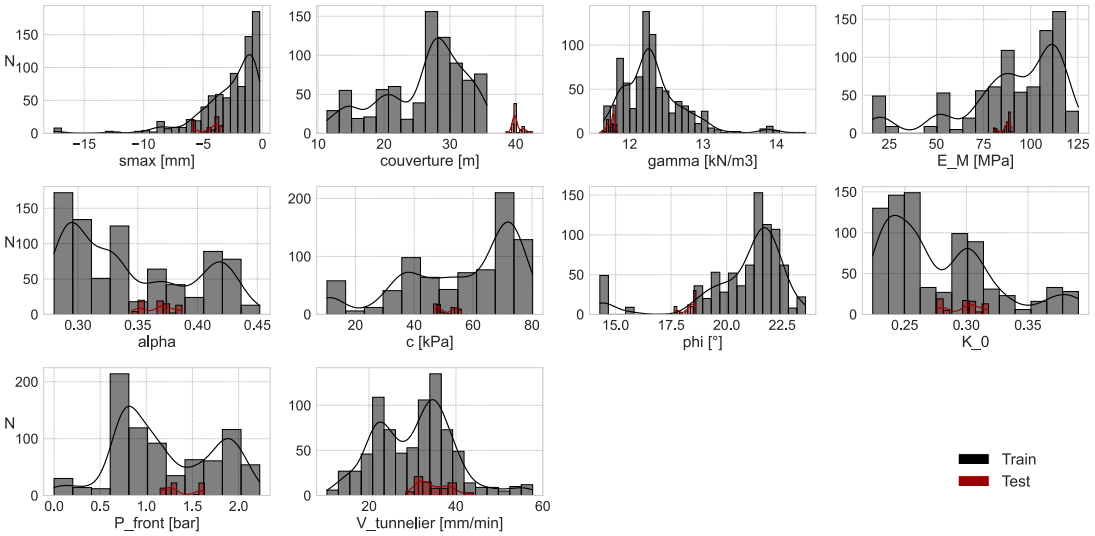


(c) Profil en long TR2 (apprentissage)

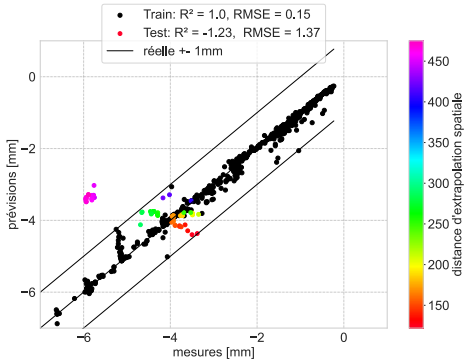


(d) Profil en long L14S2 (apprentissage)

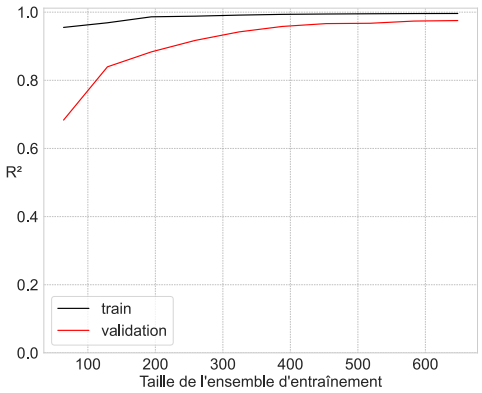
Figure 7.9. Division des données et profil en long des zones d'apprentissage et de test



(a) Distribution des données d'entraînement et de test



(b) RF



(c) Courbes d'apprentissages RF

Figure 7.10. Résultats de la prévision des tassements en entraînant sur les données 1 an après le début du creusement du TR1

stratigraphie avec des Marnes de Meudon mais sans remblai, le tassement dérive du côté sous-estimation.

7.2.2 Modèle entraîné au fur et à mesure du creusement

Enfin, nous appliquons l'approche décrite ci-dessus mais en renouvelant l'entraînement de l'algorithme tous les mois. Nous avons appliqué les tests sur les trois tronçons, et les résultats sont présentés dans les Figures 7.11, 7.12 et 7.13 pour des tests sur respectivement les tronçons TR1, TR2 et L14S2. Il convient de noter les dates de début de creusement de ces tronçons :

- TR1 : 26/02/2019
- TR2 : 26/04/2019
- L14S2 : 04/08/2019

A première vue, on remarque tout de suite que les premiers entraînement ont un nombre de points limités, ce qui est tout à fait normal puisqu'au début du creusement des tunnels nous n'avons pas encore de valeurs de s_{max}.

Concernant le TR1, on remarque que les prévisions sont bonnes, a minima, jusqu'à une distance d'extrapolation spatiale de 150 m et au bout de 5 mois de creusement (1ère figure de la Figure 7.11). On remarque que les prévisions des 2 derniers mois ont plutôt échoué. Cela s'explique par le fait que pendant cette période, le tunnelier creuse dans la zone avec une stratigraphie très différente du reste des tronçons (ce qui a déjà été discuté dans § 7.1.1).

La prévision du s_{max} sur le début du TR2 a, comme prévu, échoué. En effet, le début du TR2 est une zone avec une stratigraphie et des couvertures très variables (comme discuté dans le § 7.1.3). Néanmoins, au bout d'un an de creusement du TR2, on arrive dans la zone avec une stratigraphie plutôt homogène et le modèle réussi alors à effectuer des extrapolations spatiales sur une distance de 150 à 200 m environ (Figure 7.12).

La prévision du s_{max} sur le début du tracé de la L14S2 n'est pas évidente. En effet, le début de ce tracé a une stratigraphie différente de ce qui est observé dans les tronçons TR1 et TR2. Or, sur cette partie du tracé, nous ne disposons pas de nombreux capteurs (Figure 5.7) et donc les mesures de s_{max} sont limitées. Par conséquent, le modèle n'arrive pas à s'entraîner correctement sur cette zone ce qui explique donc les résultats médiocres de prévision.

Au bout de 9 mois de creusement, les résultats de prévision commencent à s'améliorer (Figure 7.13). Toutefois, le reste du tracé de la L14S2 passe par une zone où il y a de nombreuses carrières ce qui peut affecter la qualité de prévision de l'algorithme RF. Il est donc difficile d'établir des conclusions à propos de ce tronçon.

La conclusion principale de cette section est le fait qu'en prenant en compte les données de plusieurs tronçons, une prévision fiable peut se faire dans des zones pas encore excavés avec des caractéristiques similaires, avec une distance d'extrapolation spatiale d'environ 150 à 200 m.

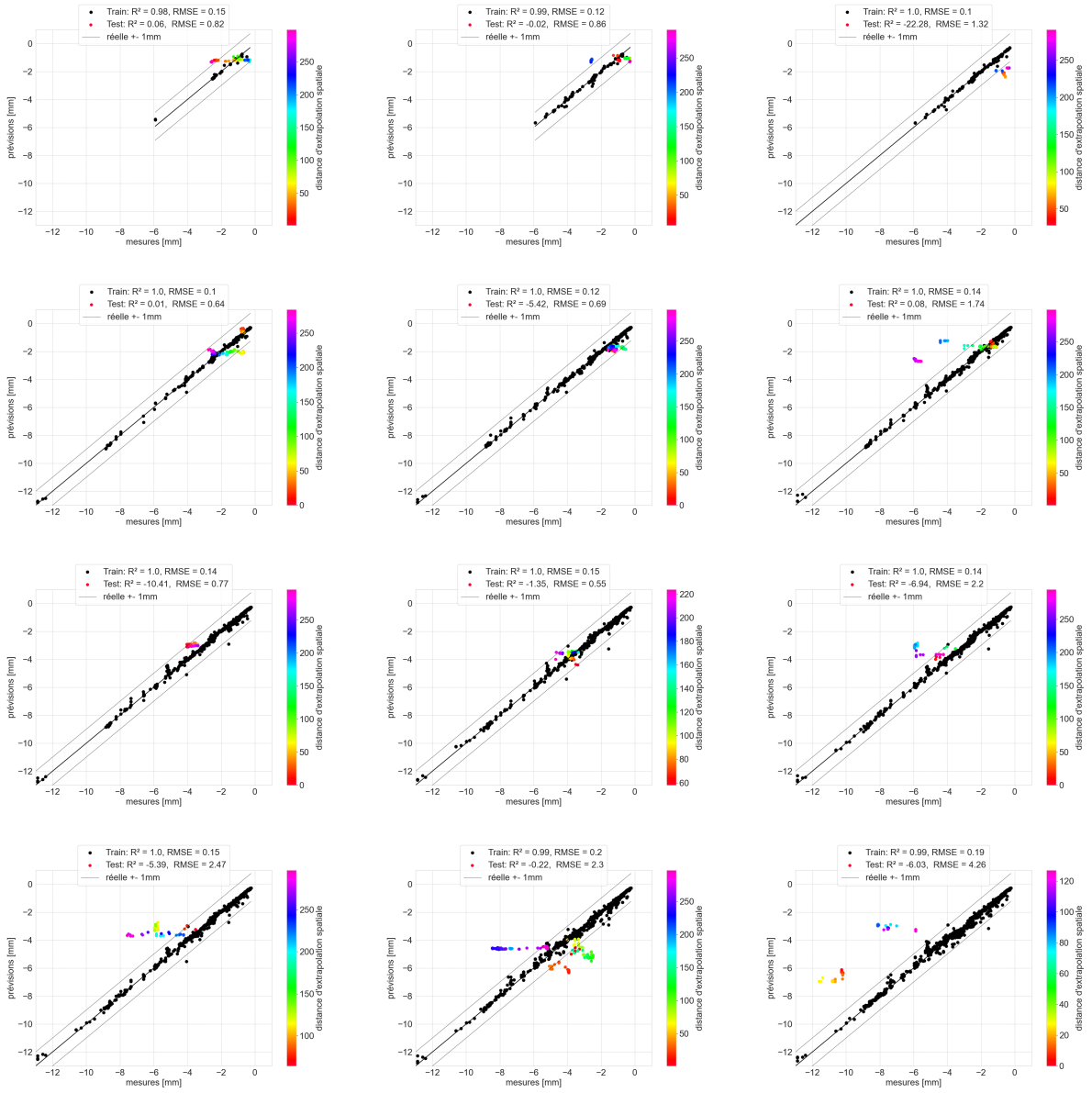


Figure 7.11. Entraînement tous les mois de RF avec prise en compte de la progression du creusement (test sur TR1).
Première date : 31/07/2019, dernière date : 30/06/2020

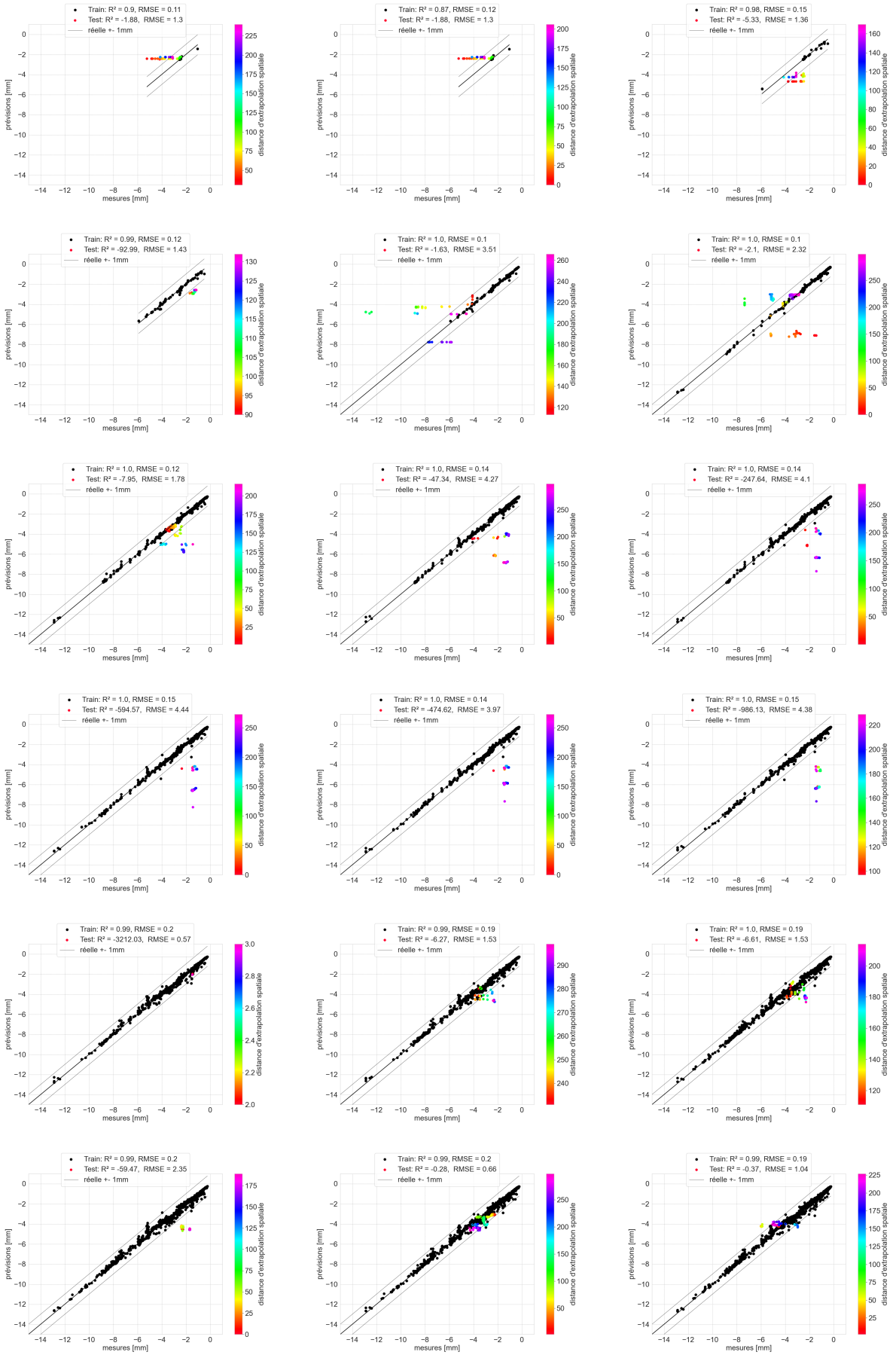


Figure 7.12. Entraînement tous les mois de RF avec prise en compte de la progression du creusement (test sur TR2).
Première date : 31/05/2019, dernière date : 31/10/2020

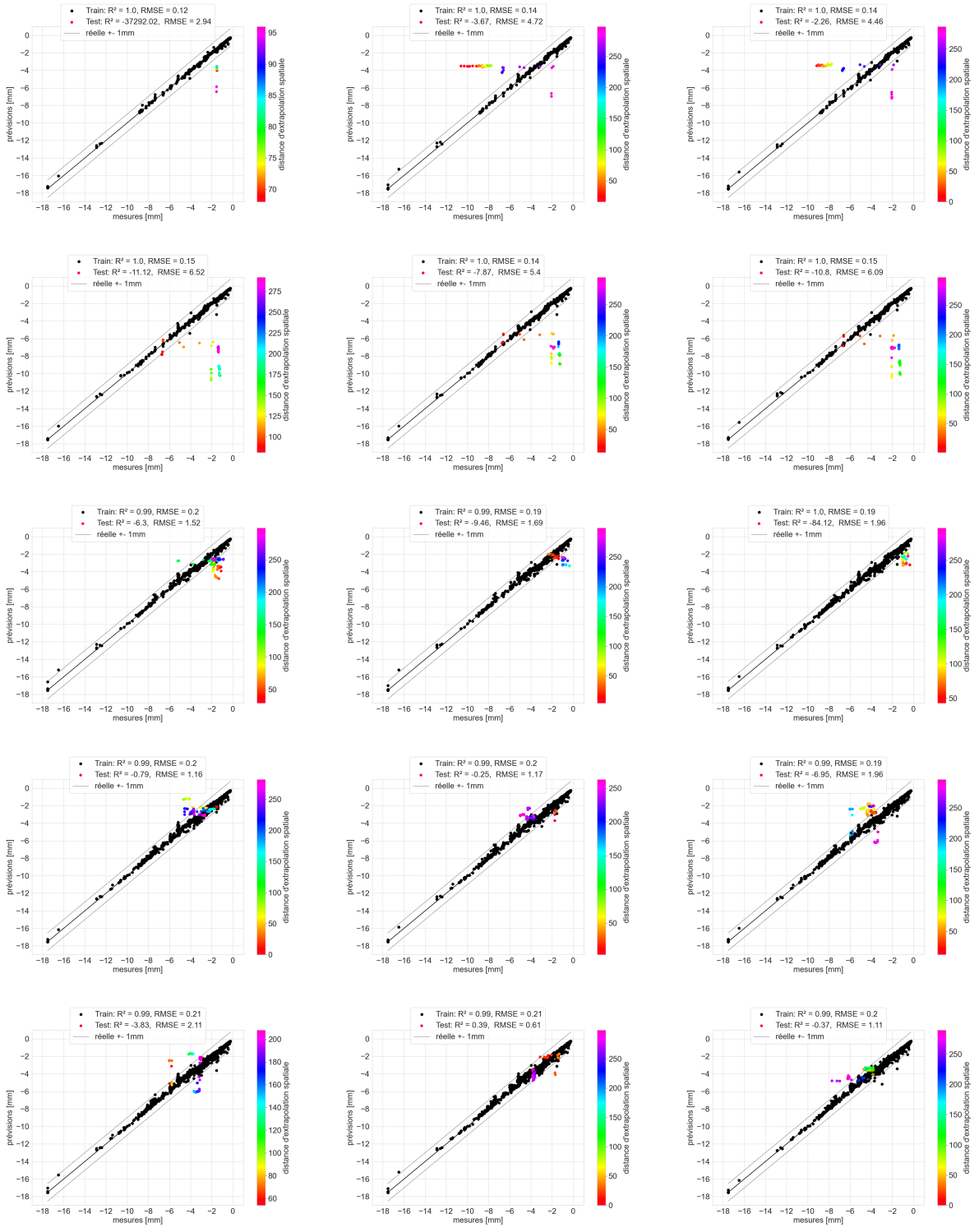


Figure 7.13. Entraînement tous les mois de RF avec prise en compte de la progression du creusement (test sur L14S2).
Première date : 30/11/2019, dernière date : 31/01/2021

Conclusion

Ce chapitre avait pour objectif de prévoir le tassement maximal à l'axe du tunnel (s_{max}) à partir de l'algorithme RF tout en tenant compte de l'aspect spatio-temporel de la progression du creusement au tunnelier. Deux parties sont étudiées.

En premier lieu, nous avons pris en compte l'aspect spatial du creusement au tunnelier en fixant une coupure entre les données d'apprentissage et de test. Cette approche nous a permis de tirer les conclusions suivantes : d'une part, les modèles obtenus à partir de RF sont incapables d'extrapolation statistique, c'est-à-dire de prévoir des tassements à partir des plages de données non-connues dans l'ensemble d'apprentissage ; d'autre part, ces modèles sont capables d'effectuer des extrapolations spatiales, c'est-à-dire de prévoir s_{max} sur une partie du tracé qui n'est pas introduite dans l'apprentissage.

On peut retenir que la distance d'extrapolation spatiale obtenue dans ce cas est d'environ 300 m, pour un entraînement sur les 1000 m précédents. Par là, on entend qu'on a des prévisions globalement correctes, avec une précision de l'ordre de 1 mm sur une gamme de valeurs de tassements qui s'étend jusqu'à environ 8 mm sur les données de test. De plus, on remarque que l'entraînement sur un tronçon supplémentaire induit une amélioration considérable : il est alors possible d'entraîner uniquement sur 500 m d'un tronçon (en plus d'un autre tronçon complet) afin de prévoir s_{max} sur les 300 m suivants. Toutefois, il faut noter que la distance d'extrapolation spatiale reste limitée en cas d'extrapolation statistique.

En second lieu, une preuve de concept POC (Proof Of Concept) sur la prévision du s_{max} en temps réel est présentée. L'idée est de ré-entraîner un modèle au fur et à mesure de la progression du creusement et donc de l'acquisition de nouvelles données. On conclut que, dans des zones homogènes, on est capable de prévoir le s_{max} 5 mois après le début du creusement et sur une distance d'extrapolation spatiale d'environ 150 m.

En somme, ce chapitre nous a permis de valider la possibilité de prévision du s_{max} en temps réel à partir d'un algorithme d'apprentissage automatique à base d'arbres (RF). Les limites de ces approches ainsi que des propositions de perspectives seront discutées dans la dernière partie.

Conclusion

Cette partie est riche en conclusions grâce aux nombreuses approches testées.

D'abord, la division aléatoire des données a permis de mettre en évidence la bonne applicabilité des algorithmes d'apprentissage automatique, notamment RF et XGBoost, pour de petits ensembles de données. La quantité de données nécessaire n'est pas aussi importante qu'on aurait pu le croire initialement à la vue des tracés utilisés pour les exercices. Cette quantité dépend bien de la variabilité des stratigraphies et paramètres rencontrés : plus la variabilité est forte, plus la quantité de données requise est importante. Cela est conforme à l'intuition qui nous dit qu'il est plus facile d'apprendre sur des données simples que sur des données complexes.

Nous avons également présenté une méthodologie claire de validation des modèles obtenus à partir des différents algorithmes, ce qui permet de comparer leurs performances. Ensuite, la régularisation et l'optimisation de DT, RF et XGBoost a permis de mieux comprendre ces modèles et ainsi limiter leur sur-apprentissage tout en optimisant leurs performances.

A partir des différentes approches adoptées, nous avons confirmé que les algorithmes à base d'arbres sont incapables d'effectuer des extrapolations statistiques tel que précisé dans l'état de l'art, mais sont prometteurs pour l'extrapolation spatiale. Pour approfondir ce sujet, nous avons appliqué une approche simple en guise d'aide à l'application en ingénierie qui prend en compte l'aspect spatio-temporel du creusement du tunnel. Nous en avons déduit que, dans des zones homogènes, il est possible de prévoir le s_{max} (tassement maximal à l'axe du tunnel) à partir de 5 mois après le début du creusement et sur une distance d'extrapolation spatiale d'environ 150 m.

Il convient de mentionner qu'une prévision des tassements maximaux à une distance donnée de l'axe du tunnel (s^*) a également été présentée. Cette approche permet de prévoir le tassement transversal sans passer par la prévision des paramètres géométriques i_y et m_y (§ 1.2.2). Les résultats sont prometteurs mais des études supplémentaires doivent être effectuées pour aboutir à des conclusions fermes.

La partie suivante propose une discussion sur les résultats obtenus et leurs limites, tout en évoquant des approches alternatives à tester en perspective de ces travaux.

Conclusions, Perspectives et Recommandations

Conclusions Générales

L'objectif de cette thèse était d'appliquer des algorithmes d'apprentissage automatique pour prévoir les tassements à l'avant du front du tunnel en temps réel, c'est-à-dire pendant le creusement au tunnelier. Les données utilisées proviennent de deux lignes de métro du Grand Paris Express, à savoir les lignes 14 Sud et 15 Sud-Ouest. Ces lignes sont creusées au tunnelier à pression de terre, ce qui nous a incité à détailler dans le Chapitre 1 le fonctionnement de ce type de machine de creusement. Ensuite, nous avons résumé la représentation mathématique du tassement ainsi que les méthodes « traditionnelles » de calculs utilisées. Les limites de ces approches ont également était présentées afin de pointer l'intérêt particulier qu'il y a d'appliquer aujourd'hui des méthodes à base d'Intelligence Artificielle. En effet, ces techniques ont l'avantage de prévoir le tassement en tout point du tracé et en temps réel, à partir d'un modèle pré-établi.

L'utilisation de ces approches n'est possible qu'après avoir acquis une bonne compréhension des algorithmes d'apprentissage automatique afin de réussir à les régulariser pour éviter le sur-apprentissage et obtenir ainsi des résultats interprétables. Le Chapitre 2 a pour vocation d'expliquer l'Intelligence Artificielle, l'apprentissage automatique (Machine Learning) et le Big Data à des ingénieurs non-spécialisés en informatique afin de les motiver et de leur donner les clés pour appliquer de telles approches dans le cadre de leurs travaux.

Cependant, pour convaincre la communauté de l'applicabilité de ces approches dans le cadre d'études d'ingénierie, il fallait trouver une méthodologie scientifique qui permette de conclure sur la confiance à attribuer aux résultats des modèles obtenus à partir d'algorithmes d'apprentissage automatique. Pour cela, de nombreux tests doivent être lancés, et cela avec une grande quantité de données puisqu'il est reconnu que les algorithmes d'apprentissage automatique sont plus performants lorsque la taille des ensembles de données augmente. Cependant, l'état de l'art aujourd'hui se base sur des corpus qui manquent souvent largement de données.

Cette thèse a l'avantage de pouvoir se baser sur une grande quantité de données brutes tirée de deux grands chantiers de tunnel. Cependant, avant de pouvoir mettre en application les algorithmes d'apprentissage automatique, un travail volumineux de traitement de ces données est nécessaire. En effet, comme nous l'avons détaillé dans la méthodologie de travail proposée, après la définition du problème, il faut collecter la donnée, la nettoyer et la stocker. Nous avons choisi dans le cadre de ce travail de créer une base de données de tunnel afin de stocker les données d'une façon permanente et sécurisée (Chapitre 4). Cette tâche s'est révélée particulièrement chronophage, mais elle constitue en soit une valeur ajoutée indéniable à ce travail du fait que ces données pourront être utilisées dans des études ultérieures et non cantonnées au cadre strict de cette thèse.

Dans le cadre de ce travail, on traite la prévision du tassement maximal à l'axe du tunnel, noté s_{max}, et du tassement maximal à une distance de l'axe du tunnel, noté s^*. Il a été proposé dans le Chapitre 5 de passer tout d'abord par le calage de l'équation de progression du tassement pour obtenir s^*. Puis, à partir de s^*, on cale l'équation du tassement transversal pour obtenir s_{max}. Cependant, pour obtenir des calages de qualité, de nombreuses approches supplémentaires de nettoyage des mesures des capteurs ont été suivies. Par exemple, nous avons appliqué un algorithme d'apprentissage automatique non-supervisé, à savoir les forêts d'isolation (IF). Ce dernier a l'avantage de supprimer des valeurs aberrantes sans perdre trop de données. Après obtention des paramètres de la cuvette de tassements, il a été possible de lancer de nombreuses études d'analyse exploratoire afin de détecter les tendances des paramètres et les corrélations entre le tassement et les différents paramètres de sol ou de pilotage du tunnelier. Par exemple, il a été possible de déterminer la proportion du tassement observé au front par rapport au tassement maximal lors du creusement au tunnelier.

L'étape suivante consiste à effectuer de nombreux exercices pour tester l'applicabilité de l'apprentissage automatique pour la prévision des tassements s_{max} et s^*.

Tout d'abord en ce qui concerne s_{max}, nous sommes d'abord partis d'une approche simple de prévision de s_{max} sans tenir compte des problématiques spatiales et temporelles, c'est-à-dire en effectuant des divisions aléatoires des données en ensemble d'apprentissage et de test. Cette application nous a permis, d'un côté, de mettre en évidence la quantité de données nécessaire pour entraîner correctement certains algorithmes d'apprentissage automatique (environ 500 observations pour notre cas d'usage) et, d'un autre côté, de comparer différents algorithmes à partir d'une méthodologie claire de validation des modèles obtenus. De plus, nous avons procédé à la régularisation et l'optimisation de DT, RF et XGBoost pour améliorer les résultats.

Ensuite, nous avons ajouté progressivement des degrés de complexité au problème en tenant compte, d'abord, de l'aspect spatial (apprentissage sur des points à l'arrière du front et test sur des points à l'avant du front) puis de l'aspect temporel (apprentissage sur des zones excavées, à une distance d'environ 100 m du front, afin de tenir compte de la disponibilité des données en temps réel sur le chantier).

Ces études nous ont permis de tirer les conclusions suivantes : d'une part, nous avons confirmé que les algorithmes à base d'arbres sont incapables d'effectuer des extrapolations statistiques, c'est-à-dire de prévoir sur des zones où les paramètres d'entrée sont peu ou pas connus par l'algorithme lors de son apprentissage ; d'autre part, nous avons déduit que dans des zones homogènes, on est capable de prévoir le s_{max} (tassement maximal à l'axe du tunnel) à partir de 5 mois après le début du creusement et sur une distance d'extrapolation spatiale d'environ 150 m.

Ensuite en ce qui concerne s^*, nous avons également présenté dans cette thèse la prévision des tassements maximaux à une distance de l'axe du tunnel (s^*) qui permettrait de

prévoir le tassement transversal sans passer par la prévision des paramètres géométriques i_y et m_y (§ 1.2.2). Cependant, des études supplémentaires sont nécessaires pour aboutir à des conclusions fermes. Les résultats obtenus pour s^* sont globalement similaires à ceux obtenus pour s_{max} même s'ils sont légèrement moins satisfaisants.

Cette thèse, effectuée dans le cadre d'une convention CIFRE, a eu le privilège de mixer la recherche avec les aspects pratiques de la profession. Des perspectives d'études s'ouvrent tels que l'application d'approches plus adaptées à la complexité des problèmes spatio-temporels. Nous proposons dans la suite une série de discussions à ce sujet et bien d'autres en plus de quelques recommandations d'ordre pratique à propos des données et de la méthodologie d'application des algorithmes d'apprentissage automatique.

Discussions et Perspectives

Les conclusions présentées dans la partie précédente sont l'aboutissement de nombreux essais effectués au préalable. En effet, de nombreuses idées ont émergé lors de nos travaux, notamment sur le choix de la variable cible et de la méthode de réduction des dimensions des paramètres géologiques et géotechniques. De surcroît, des questions se sont posées vis-à-vis des résultats obtenus à partir du calage des paramètres de tassement. Ces idées et réflexions sont présentées sous forme de discussions et perspectives dans ce chapitre avec comme fil conducteur l'envie de proposer de nouvelles pistes d'amélioration et d'approfondissement du sujet, qu'elles soit techniques ou pratiques.

A propos de l'apprentissage automatique

Dans cette partie, nous présentons une discussion sur les limites des modèles d'apprentissage automatique obtenus (§ 7.2) pour ensuite proposer des approches alternatives en guise de perspectives.

Limites des modèles proposés

Le dernier modèle proposé dans le § 7.2 a l'avantage de prendre en compte l'aspect spatio-temporel du creusement, c'est-à-dire que c'est un modèle capable de s'entraîner au fur et à mesure du creusement au tunnelier et de prévoir les tassements en temps réel, pour des zones pas encore creusées, à l'avant du front du tunnel. Cependant, ces modèles ne sont pas sans limitation.

Tout d'abord, comme nous l'avons déjà dit, ces modèles ne sont fonctionnels que dans des zones au profil géotechnique homogène, ou, a minima, des zones similaires à d'autres zones prises en compte dans l'apprentissage du modèle. Cela est dû à leur incapacité à gérer l'extrapolation statistique (§ 6.2 et Chapitre 7).

Ensuite, à partir des résultats des analyses d'importance des caractéristiques, on trouve que ces modèles sont davantage influencés par les paramètres combinés de sol que par les paramètres de pilotage du tunnelier (§ 6.2). Il n'est donc pas possible, à ce stade, de formuler des prescriptions claires d'actions préventives à mener sur le chantier en cas de prévisions de tassements qui rendraient vulnérables des structures en surface. Il faut alors compter sur le jugement des ingénieurs pour cette tâche. L'action la plus commune serait l'augmentation de la pression de confinement, au risque cependant de diminuer les cadences d'avancement.

Enfin, dans un cas réel, la qualité des prévisions effectuées ne peut être garantie instantanément. En effet, il faut attendre que les tassements maximaux soient obtenus dans la zone de prévision, soit à minima 75 m après le passage du front. Pour aller plus loin, il faudrait donc prévoir le tassement en temps réel, et non pas uniquement le tassement maximal. Cela permettrait à la fois de prévoir l'évolution des tassements dans

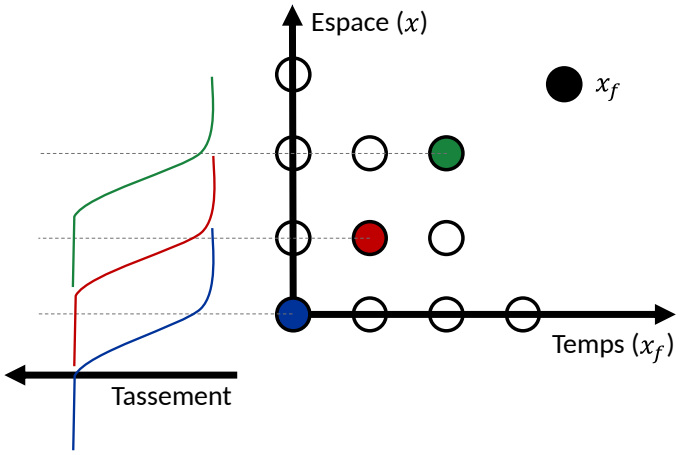


Figure 7.1. Explication de l'application de l'approche bayésienne pour la prévision des tassements en tout point de l'espace en fonction de l'avancement du creusement

la zone du creusement (notamment la pente longitudinale transitoire qui peut avoir un impact sur les structures de surface), mais également de ne pas être dans l'obligation d'exclure les données de cette zone de transition pour l'apprentissage. De plus, une telle approche permettrait de profiter de toutes les mesures des capteurs sans passer par le calage. Pour rappel, notre base de données contient aujourd'hui 13 141 328 mesures de tassements pour des distances au front comprises entre 150 m avant et 500 m après le passage du tunnelier. Nous présenterons dans ce qui suit des approches qui pourront répondre à ce besoin.

Prévision du tassement instantané

Dans cette partie, nous proposons des approches qui pourraient être capables de prévoir le tassement en temps réel et en tout point de l'axe du tunnel, et donc la courbe de progression du tassement avec l'avancement du tunnel (§ 1.2.2). Pour résoudre un tel problème, on peut utiliser des algorithmes intégrant l'effet temporel, tel que les LSTM.

D'autre part, on peut tenter de relier les aspects spatiaux et temporels du problème en utilisant des modèles adaptés tels que les réseaux bayésiens (Figure 7.1). Pour cela, on peut incorporer au modèle des variables qui capturent l'information spatiale, telles que la position du point de mesure et l'information sur la nature des sols, et des variables qui capturent l'information temporelle, telle que la position du front du tunnel. Les réseaux bayésiens peuvent ensuite utiliser ces variables pour modéliser la dépendance spatiale et temporelle dans les données, en estimant les probabilités conditionnelles entre les différentes variables. Ces modèles peuvent ensuite être utilisés pour prévoir la probabilité de tassement à une position donnée et à une date donnée, en prenant en compte l'information spatiale et temporelle pertinente.

Une autre application possible est la prévision des incréments de tassement. Ces derniers sont la différence de tassement observé en un point donné pour un certain pas

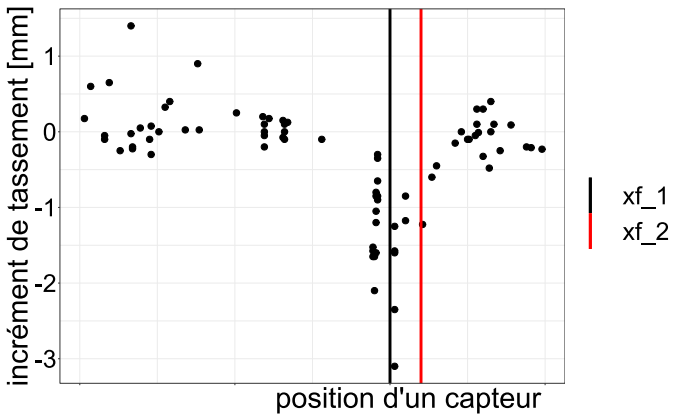


Figure 7.2. Incrément de tassement observé pour un creusement de 20 m. Les mesures affichées sont obtenues par des capteurs à une distance inférieure à 20 m de l'axe du tunnel. Légende : x_f

d'avancement du creusement. Un incrément de tassement pour un creusement de 20 m est proposé dans la Figure 7.2. On remarque que le plus grand tassement induit par ce pas de creusement est observé au front du tunnel. Théoriquement, on peut caler une équation gaussienne sur les incréments de tassement. En pratique, cela n'est pas évident et nécessite des travaux de nettoyage des mesures comme présentés pour le calage des équations de tassement transversal et de progression du tassement (§ 5.2). De plus, il convient de noter que l'exemple proposé dans cette figure est un cas idéal par rapport à la majorité.

A propos de la modélisation du sol

Le problème des sols est le besoin d'un grand nombre de paramètres pour les représenter correctement. Dans cette thèse, nous avons adapté une approche de combinaison des paramètres de sols (Chen et al., 2019a) en prenant en compte la couverture du tunnel ainsi que la position et l'épaisseur des différentes couches. Dans ce qui suit, nous proposons des approches alternatives.

Auto-encodeurs

D'après les résultats des études proposées dans cette thèse, on déduit que les paramètres combinés des sols ont perdu leur signification physique et, malgré cela, les algorithmes d'apprentissage automatique ont été capables de développer une compréhension de ces paramètres. Par conséquent, il est envisageable d'utiliser des paramètres mathématiques dépourvus de signification physique directe, tout en sachant que ces paramètres pourraient être utiles pour la modélisation et à l'analyse des données. Pour obtenir ces paramètres, nous avons testé une autre approche qui a été développée dans le cadre du stage de Selmane Lebdaoui à Setec Terrasol : celle des auto-encodeurs (Lebdaoui, 2022). Ces derniers sont un ensemble de deux réseaux de neurones : le premier transforme les

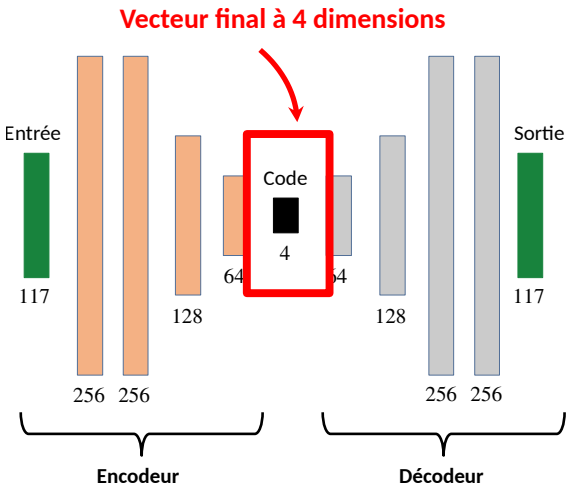


Figure 7.3. Architecture de l'auto-encodeur utilisé

paramètres d'entrée de grande dimension en un seul vecteur de taille à définir (4 dans notre cas, Figure 7.3) alors que le deuxième décode le vecteur obtenu par le premier réseau pour obtenir les paramètres d'entrée initiaux. En d'autres termes, le premier réseau code et le second décode afin d'évaluer la pertinence du vecteur final. Il convient de noter que les paramètres d'entrée utilisés dans le cadre de cette étude sont : γ [kN/m^3], E_M [MPa], α, K_0, c [kPa] et φ [°] ainsi que C [m], h et e. Au final, on obtient 4 caractéristiques représentatives de la stratigraphie à introduire au modèle de prévision de tassement au lieu de 6 caractéristiques avec la méthode de combinaison adoptée dans cette thèse.

Les premiers résultats ont montré que cette approche est prometteuse avec des performances seulement légèrement inférieures à celles obtenues avec les paramètres de sols combinés. Ces résultats ne sont pas présentés dans le cadre de cette thèse puisqu'ils ont été mené sur des données légèrement différentes (vis-à-vis du nettoyage et du calage des courbes de tassement), ce qui ne permet plus de comparer ces résultats avec ceux présentés dans ce manuscrit. Dans tous les cas, la méthode de combinaison des paramètres de sol pour la réduction des dimensions reste la méthode à privilégier compte tenu de sa rapidité d'exécution. En effet, la méthode des auto-encodeurs consomme beaucoup d'énergie et de temps et est moins rapide puisqu'il faut déterminer l'architecture optimale des réseaux de neurones (ici établie à partir d'essais / erreurs) et les entraîner avec une quantité suffisante de données. Une axe qui reste à approfondir avec l'utilisation des auto-encodeurs consiste à s'assurer que cette approche est capable de remonter à la bonne valeur des paramètres géologiques et géotechniques initiaux.

Approche simplifiée

D'après les résultats obtenus dans le § 6.2.2, les modèles n'ont finalement pas besoin d'une grande quantité de données pour s'entraîner. Compte tenu de la taille de notre ensemble de données, on peut se permettre alors d'augmenter le nombre de caractéristiques à

prendre en compte, avec évidemment une augmentation de la proportion de l'ensemble d'apprentissage par rapport à celle de l'ensemble de test. Il serait également intéressant de tester d'autres combinaisons de paramètres comme par exemple en introduisant uniquement les paramètres des sols au front du tunnel pour voir si c'est le front qui domine toute la réponse du massif. De plus, on pourrait prendre en compte la présence des nappes (non prises en compte dans les exercices présentés) et des carrières en tant que variables catégorielles (0 et 1 en fonction de la présence ou non de nappes/carrières).

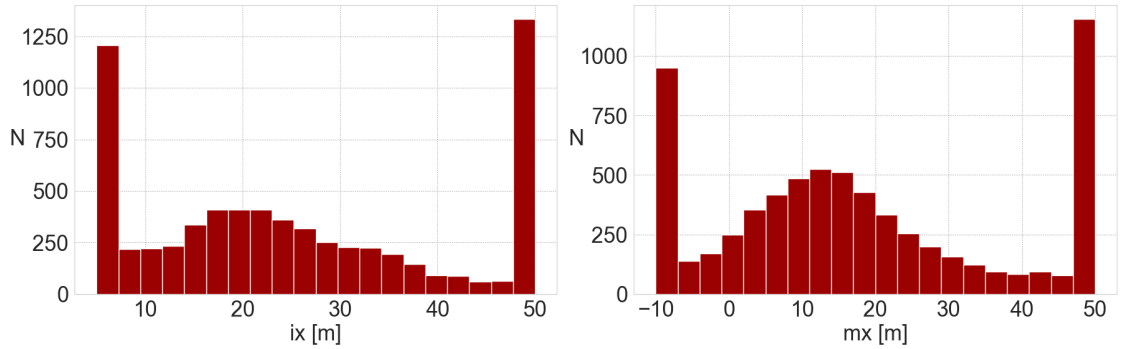
Il convient de noter également que toutes ces approches se basent sur des paramètres géotechniques retravaillés par les géotechniciens en fonction de leur expertise. Cependant, il serait intéressant de tester des algorithmes d'apprentissage automatique avec des données brutes de sol, provenant directement d'essais in-situ ou de laboratoire, sans intervention humaine.

A propos du calage des paramètres des équations de tassement

Nous avons présenté dans le § 5.2 la méthode de calage des équations temporelles et transversales des tassements. Les résultats des distributions des paramètres obtenus sont discutés dans le § 5.3.1. Une suite à cette discussion est proposée ici.

Influence des bornes

Les effets de bords observés sur les paramètres i_x et m_x indiquent clairement que le calage peut être optimisé. Nous avons donc relancé le calage des équations sur les courbes mais cette fois-ci en limitant i_x entre 5 m et 50 m (contre 5 m et 100 m auparavant) tandis que m_x est libéré avec des bornes entre -10 m et 50 m (contre -10 m et 30 m avant). Les résultats des distributions de i_x et m_x sont présentés dans la Figure 7.4. On observe qu'à partir d'environ 30 m, les valeurs de m_x ne sont qu'une longue traîne de distribution ce qui confirme notre choix initial. Avec le même raisonnement, on peut admettre une limite de 40 m pour i_x.



(a) i_x (b) m_x

Figure 7.4. Distribution des paramètres i_x et m_x avec les nouvelles limites

Pour cette problématique spécifique, la valeur d'intérêt est s^*. Il convient donc d'observer l'effet de ces nouvelles limites de calage sur ce paramètre. Pour cela, nous comparons les valeurs de s^* issues du calage avec les anciennes bornes avec celles obtenues avec les nouvelles bornes de i_x et m_x. Le résultat est présenté dans la Figure 7.5. On constate qu'il y a 106 capteurs dont les nouvelles limites impliquent une différence de s^* supérieure à 1 mm. La différence maximale observée est de 8 mm, la moyenne des différences est de 0.02 mm et la médiane est de 10^{-7} mm. On en conclut que les valeurs des limites des paramètres i_x et m_x n'a pas une influence considérable sur les valeurs de s^*.

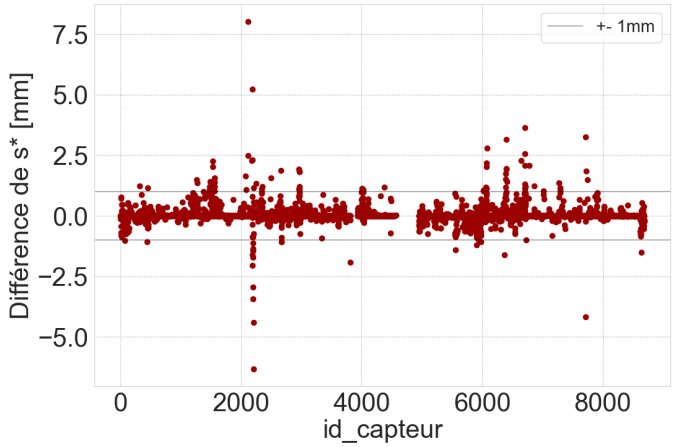


Figure 7.5. Différence entre les valeurs de s^* calées avec les deux jeux de bornes pour i_x et m_x pour l'ensemble des capteurs

Qualité ou quantité ?

Dans cette étude, nous avons présenté de nombreuses techniques de nettoyage des capteurs et des mesures de tassement associées telles que le lissage des courbes avec des médianes mobiles ou des moyennes sur des distances, l'élimination des valeurs aberrantes avec l'algorithme des forêts d'isolation (IF), le calage des paramètres des équations avec la méthode des moindres carrés et l'évaluation du calage avec le coefficient de détermination R^2. L'efficacité de chacune des techniques ainsi que leur optimisation peuvent faire l'objet de longues discussions. Au final, il convient de retenir qu'il n'y a pas de méthode idéale : ce qui fonctionne sur un ensemble de capteurs peut facilement ne pas être convenable sur d'autres. Dans notre cas, nous avons privilégié la quantité de données sur la qualité afin d'avoir le plus grand jeu de données possible pour l'entraînement des algorithmes. Néanmoins, ce choix est discutable à l'égard des conclusions présentées dans le § 6.2.2, qui montrent qu'un jeu de données composé de 800 mesures environ est suffisant pour entraîner des algorithmes ensemblistes tels que RF et XGBoost. Par conséquent, nous pouvons recommander aux futures études qui seraient sur ce même sujet de privilégier la qualité sur la quantité, ou en tous les cas, de rechercher un bon équilibre entre quantité et qualité. Pour obtenir la meilleure qualité possible sur le calage, on recommande de limiter

le R^2 à une valeur plus stricte. A titre d'exemple, la Figure 5.12 montre la différence qui existe entre différents R^2. En complément, on peut approfondir la réflexion sur les limites des paramètres lors du calage tout en limitant le nombre d'itérations (à travers l'argument *maxfev*) et on peut également utiliser des médianes sur les distances, ce qui permet de réduire le bruit des mesures.

RECOMMANDATIONS

Recommandations sur les données

A propos du traitement des données

Nous présentons dans ce qui suit une liste non exhaustive des gammes de problèmes rencontrés dans le cadre spécifique de l'automatisation des prévisions de tassements des tunnels.

Propriété et confidentialité de la donnée (data confidentiality) : la donnée (de monitoring ou de reconnaissance géotechnique) est acquise par une entreprise, un laboratoire, etc. mais elle est la propriété du Maître d'Ouvrage. Par conséquent, la donnée brute, nécessaire à des analyses poussées via des modèles d'apprentissage, est le plus souvent difficile à acquérir par un tiers chargé du suivi et des analyses (comme le Maître d'Œuvre). Il convient de préciser dans les documents du marché les obligations de chaque entreprise du point de vue des données acquises pour éviter les difficultés d'ordre contractuel lors des demandes d'extraction et de restitution finale.

Diversité des fournisseurs de données : les instrumentations chantier sont habituellement effectuées par des sous-traitants choisis par les entreprises d'exécution. On peut alors se retrouver sur un certain projet avec plusieurs fournisseurs de données et par conséquent plusieurs sources de données (base de données, sites web, applications, etc.). Cette diversité de sources et d'architectures des données complique la tâche d'extraction. Là encore, il pourrait être souhaitable d'imposer aux fournisseurs l'intégration des données dans une base préalablement construite par celui qui en fait les analyses. Une autre solution peut-être de demander systématiquement un accès aux données via une API (Application Programming Interface) documentée qui permette d'y accéder sans intermédiaire.

Extraction des données (data extraction) : en pratique néanmoins, la donnée est rarement disponible directement dans une base de données. Un travail d'extraction depuis une plateforme source est la plupart du temps nécessaire, que ce soit en passant par des sites web ou des applications de bureau, etc. Cette tâche nécessite des connaissances spécifiques comme le web scraping (processus d'extraction de contenu et de données de sites web à l'aide de logiciels ou d'algorithmes en langage Python, par exemple) ou la décompilation de fichiers de stockage tampon pour certaines applications de bureau. Ces procédés détournent la fonction première de ces applications qui se bornent à une consultation des données et des analyses préformatées. Or, le fournisseur de donnée ne peut prévoir toutes les applications qui peuvent être faites avec les données qu'il collecte. C'est la raison pour laquelle on revient à nouveau sur le besoin de disposer d'accès directs aux données brutes.

Données silotées (data silos) : les données silotées sont des données détenues par un groupe mais qui ne sont pas facilement ou entièrement accessibles par d'autres groupes de la même organisation (Talend, 2022b). Pour capitaliser sur la donnée, il faut la centraliser et la rendre accessible à tous de façon uniformisée.

Diversité des formats : les données sont actuellement rarement numérisées et souvent fournies sous multiples formats (base de données, fichiers tabulaires (CSV, Excel), pdf, images, vidéos, audio, etc.) qui peuvent être différents du format nécessaire pour les études a posteriori. Il existe diverses techniques pour obtenir les données utiles à partir d'un format initial. Par exemple, la reconnaissance optique de caractères (OCR Optichal Character Recognition) permet d'extraire du texte à partir d'images de textes imprimés. Néanmoins, la tendance à stocker les données dans des bases structurées se développe et tend à faire disparaître cette problématique. Il faut donc encourager cette pratique soit par l'incitation, soit l'imposer contractuellement dans les marchés.

Données non structurées (unstructured data) : les données structurées sont des données organisées de manière à ce qu'elles soient notamment faciles à utiliser par des algorithmes d'apprentissage automatique (Talend, 2022a). Elles suivent souvent un format prédéfini et peuvent être stockées dans des bases de données ou des fichiers tabulaires, comme des fichiers CSV ou des tables SQL. Les données structurées sont généralement des données quantitatives, c'est-à-dire des données numériques ou des données catégorielles. Contrairement aux données structurées, les données non structurées ne suivent pas de format prédéfini et sont plus difficiles à utiliser pour les études a posteriori. Des exemples de données non structurées sont des audios, images, vidéos, etc. qui ont besoin d'un pré-traitement pour extraire la donnée.

Données non unifiées : il convient de veiller à l'unification des données en termes de langue, de vocabulaire, de définition, de symbole, d'unités de mesure, etc. Toutefois, il n'existe pas de consensus sur ce sujet pour harmoniser les pratiques.

Données sales (dirty data) : par définition, les données sales sont des données inexactes, incomplètes ou incohérentes. Ce sont, par exemple, des données qui contiennent des erreurs telles que des fautes d'orthographe ou de ponctuation, des données incorrectes, des données périmées, des données dupliquées, des données aberrantes (outliers), etc. Ces données nécessitent alors un nettoyage avant leur utilisation. C'est le cas des données de chantier. Les erreurs ou le bruit proviennent notamment des conditions météorologiques, du niveau d'intervention humaine, des erreurs de mesure des instrumentations, etc. Le nettoyage des données améliore la qualité du jeu de données (dataset) ce qui apporte de la confiance dans l'ensemble des données et la fiabilité des résultats d'analyse de données (Klein et Lehner, 2009 ; Krishnan et al., 2016).

Interopérabilité des données (interoperability) : c'est la capacité de transmettre l'information d'un système (logiciel, instrumentations ou machine connectée sur chan-

tier, etc.) à un autre sans opération manuelle. Le but est donc de comprendre la donnée, de l'exploiter et de la réutiliser autant que possible.

Contextualisation de la donnée : la donnée géotechnique en particulier doit être contextualisée afin d'être correctement interprétée : quelle procédure de mesure a été employée ? comment l'échantillon a-t-il été prélevé ? etc. La provenance et la traçabilité de la donnée sont alors indispensables dans ce cadre (Beaufils et Serieys, 2022).

Données éparses (sparse data) : Les données éparses sont des données qui contiennent beaucoup de valeurs manquantes ou nulles. Par exemple, l'information sur la nature du sol n'est connue que dans les lieux de sondage.

A propos des bases de données

Les avantages d'une base de données sont présentés dans le § 2.1.2. En se basant sur le retour d'expérience issu de la base de données construite et exploitée dans le cadre de cette thèse, nous proposons dans cette partie quelques recommandations d'ordre pratique pour la construction d'une base de données par un ingénieur non spécialisé en informatique.

La première étape est le regroupement des différentes catégories de données en des tables (qui seront peut être modifiées par la suite). Par exemple, dans notre cas, nous avons identifié quatre catégories : les généralités (projet, position, etc.), les paramètres de sol, les mesures de déformation, les paramètres de creusement. Il est d'usage d'ajouter également des tables d'assistance qui contiennent les abréviations, symboles, unités, etc.

Ensuite, il faut concevoir l'architecture de la base de données. Pour cela, on trace des croquis de diagrammes entité-relation afin d'identifier les relations présentes entre nos différentes tables. L'architecture doit également être conçue de sorte que la base de données reste généralisable (c'est-à-dire quelle puisse s'adapter à différentes tailles de données, scalable) . Par exemple, une base de données de creusement de tunnel au tunnelier aura vocation à accepter des données de creusement de nouvelles lignes. La conception de l'architecture est toujours l'étape la plus chronophage car il faut concilier la technique et le domaine d'application. De plus, il faut se mettre d'accord sur le format de distribution de la donnée. Cela permet, d'un côté, d'éviter à l'utilisateur final de composer des requêtes complexes pour obtenir l'information qui l'intéresse, et d'un autre côté, d'assurer l'intéropérabilité de la donnée pour l'exploiter et la réutiliser autant que de besoin. Pour arriver à ce niveau de simplicité d'emploi dans la base de données, il faudra accepter d'effectuer des transformations et modifications au préalable telles que des jointures de table et la création de colonnes qui serviront de clés primaires et secondaires. Dans notre cas, on peut citer la création et la propagation de la clé ***id_anneau*** dans l'ensemble des catégories de données, ce qui a énormément facilité l'exploitation de la base de données par la suite.

Ce travail permet également d'éviter l'utilisation de plusieurs colonnes combinées en

tant que clés primaires et secondaires, facilitant ainsi d'autant les requêtes. De plus, il est fortement recommandé de suivre un critère uniforme pour la sélection des clés primaires dans une base de données relationnelle. Une méthode courante consiste à utiliser des attributs de type *SERIAL* pour les colonnes de clé primaire, qui génèrent automatiquement des valeurs numériques uniques et croissantes pour chaque enregistrement dans la table. Cette approche présente l'avantage d'utiliser peu d'espace de stockage dans la base de données. D'ailleurs, pour cette raison même, il est important de choisir judicieusement le type d'attribut convenable pour chacune des colonnes sans abuser de l'espace alloué. Par exemple, pour les colonnes contenant des chaînes de caractères, il est préférable de sélectionner une taille appropriée pour les chaînes qui seront stockées dans ces colonnes plutôt que de choisir la taille maximale disponible. Cela permet également d'accélérer les requêtes.

Avant la création d'une telle base, il convient de réfléchir à la nomenclature des tables et des colonnes de façon à ce que cela soit le plus explicite possible. De plus, il faut adopter certaines conventions, par exemple mettre une majuscule pour le nom des tables mais pas pour le nom des colonnes, utiliser des « _ » pour éviter les espaces, ou bien choisir des noms en anglais pour contourner les caractères spéciaux, etc.

La création de la base de données doit à notre sens se faire à travers l'écriture explicite des scripts SQL qui permettent de générer l'ensemble de sa structure, et de bannir sa construction manuelle via un logiciel d'administration et de requêtage de base de données (SQL client software application, comme *DBeaver*). Cela permet de garder une trace et une reproductibilité de la création de la base de données en cas de problèmes. Dans notre cas, nous avons régénéré la base à de nombreuses reprises, pour en affiner l'architecture et fluidifier son fonctionnement. Passer par des étapes manuelles aurait été rédhibitoire, car long et peu fiable. De même, toute modification ultérieure de cette base de données doit être enregistrée (conservation des lignes de commandes ayant produit ces effets) de manière à retrouver leur trace et pouvoir reconstruire la base à différentes étapes.

La base de données sera certainement à disposition de plusieurs utilisateurs. Il convient donc de la stocker sur un serveur en ligne. Il existe pour cela de nombreux fournisseurs de solutions d'infrastructures cloud clés en main, déployables en quelques clics, et peu coûteuses. Cela facilite grandement l'accès multi-utilisateurs aux données mais aussi et surtout la sécurité et la robustesse de la base vis-à-vis de ces accès. Ces solutions sont enfin généralement fournies avec des systèmes de sauvegardes régulières automatisées.

Méthodologie d'implémentation d'un modèle d'apprentissage automatique

Cette thèse a présenté l'application des algorithmes d'apprentissage automatique pour la prévision du tassement. La méthodologie suivie pour ce travail de thèse peut être

généralisée à l'application de n'importe quel cas d'étude dans le domaine du génie civil. Nous présentons dans ce qui suit les premières réflexions à faire avant de lancer un exercice d'apprentissage automatique et nous résumons ensuite les étapes bien détaillée tout au long de ce manuscrit en un logigramme (§ 7.1).

La première étape est une vue d'ensemble du problème. En effet, avant de se lancer dans la mise en place d'un algorithme d'apprentissage automatique, une série de question doit être posées pour bien cadrer le problème.

1. Échanger avec les experts métiers pour comprendre le contexte et les enjeux.
2. Définir la problématique et l'objectif professionnel.
3. Cibler le but ultime : comment ce modèle sera utilisé pour en tirer partie.
4. Se renseigner sur l'existence de solutions existantes pour résoudre le problème ainsi que sur leurs limitations éventuelles. Cela permet de fournir des valeurs de référence, d'inspirer une façon de résoudre le problème et de lister les apports des algorithmes d'apprentissage automatique.
5. Choisir la ou les variable(s) cible(s) (target) en fonction des réponses obtenues par les experts métiers.
6. Rechercher les paramètres ayant une influence sur la (les) variable(s) cible(s). Ces paramètres aident à choisir les données nécessaires pour la suite de l'étude.
7. La disponibilité des données : a-t-on des données à notre disposition ? Quelle est la source des données : plateforme web, application de bureau, fichiers à plats, documentés ou non, base de données ? Faut-il faire des travaux d'extraction et de traitement des données ? Si la quantité de données n'est pas suffisante, peut-on générer de la donnée, par exemple à partir des méthodes de calcul usuelle ?

L'étape suivante au cadrage du problème métier est la conception du système d'apprentissage automatique. Pour cela, il convient de se référer à la Figure 2.7 pour caractériser le système : mode de consommation de la donnée, mode de généralisation ainsi que le mode d'apprentissage. Ensuite, la Figure 2.11 propose un panel des tâches les plus usuelles en fonction du mode d'apprentissage avec quelques exemples d'application et d'algorithmes. La conception du système permet un choix rigoureux des algorithmes d'apprentissage automatique les plus convenables pour le cas d'étude en question.

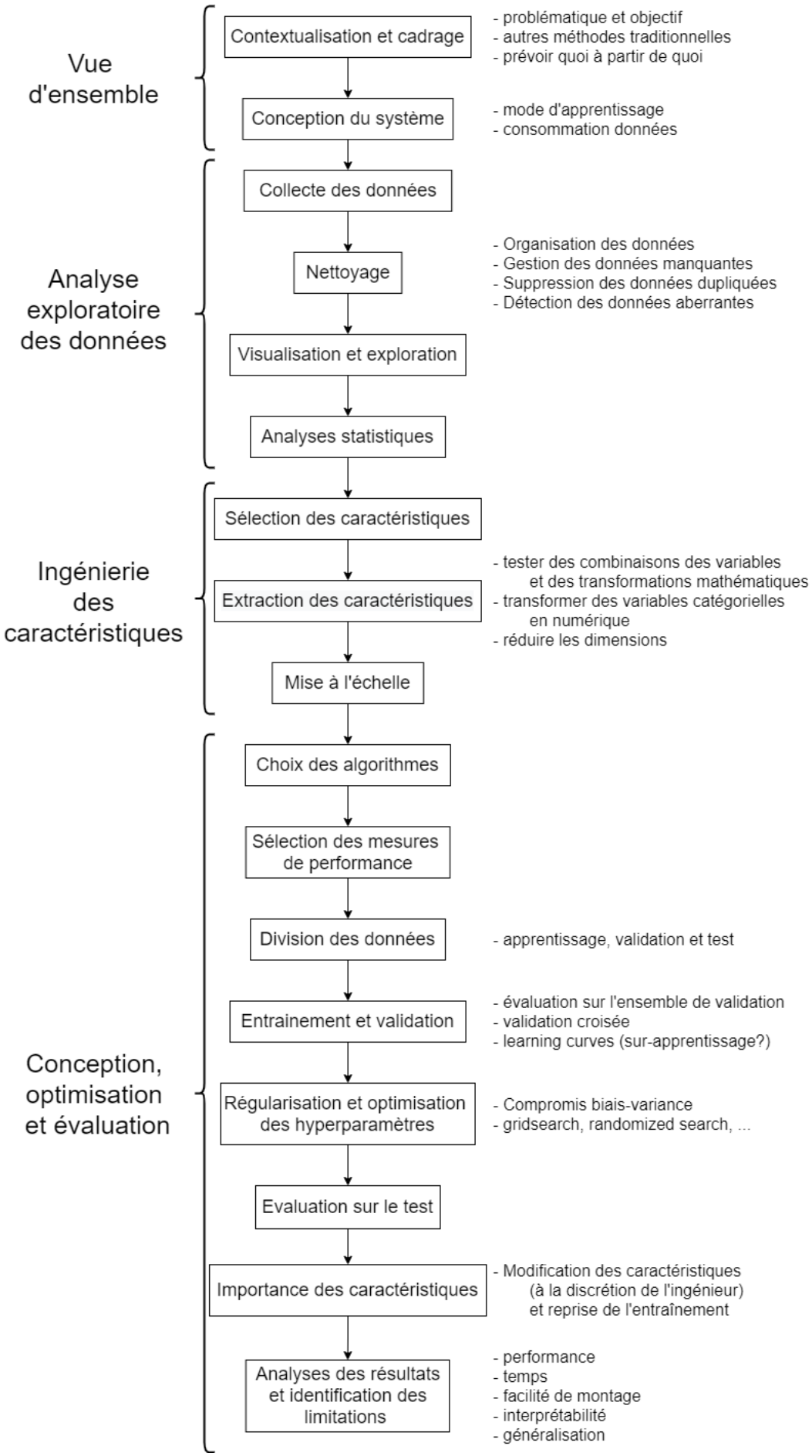


Figure 7.1. Logigramme des étapes à suivre pour une application d'apprentissage automatique

Bibliographie

AFTES (1995). *Tassements liés au creusement des ouvrages en souterrain (GT16R1F1)*. Rapp. tech. 132, p. 1-23 (cf. p. 19, 20).

AFTES (2002). *La méthode convergence-confinement (GT7R6F1)*. Rapp. tech. (cf. p. 27).

AFTES (2019). *État de l'art concernant les évolutions des tunneliers et de leurs capacités de 2000 à 2019 (Groupe de Travail n°4 - GT4R6F1)*. Rapp. tech. (cf. p. 10, 11).

AGS (2022). *Electronic Transfer of Geotechnical and Geoenvironmental Data-AGS 3.1*. Rapp. tech. 4.1.1. Association of Geotechnical et Geoenvironmental Specialists (cf. p. 39).

Ahangari, K., S. R. Moeinossadat et D. Behnia (2015). « Estimation of tunnelling-induced settlement by modern intelligent methods ». *Soils and Foundations* 55.4, p. 737-748. DOI : 10.1016/j.sandf.2015.06.006 (cf. p. 65, 67, 70, 74).

Alexandre, L. (2017). *La guerre des Intelligences*. Jean-Claud, p. 339 (cf. p. 35).

Aristaghes, P. et P. Autuori (2001). « Calcul des tunnels au tunnelier ». *Revue Française de Géotechnique* 97, p. 31-40. DOI : 10.1051/geotech/2001097031 (cf. p. 27).

Attewell, P. et J. Woodman (1982). « Predicting the dynamics of ground settlement and its derivatives caused by tunnelling in soil ». *Ground Engineering* (cf. p. 19, 22-25, 124).

Baghbani, A., T. Choudhury, S. Costa et J. Reiner (2022). « Application of artificial intelligence in geotechnical engineering : A state-of-the-art review ». *Earth-Science Reviews* 228.March, p. 103991. DOI : 10.1016/j.earscirev.2022.103991 (cf. p. 62, 63).

Beaufils, M. et A. Serieys (2022). « Gestion des Données et nouvel environnement numérique en Géotechnique ». *Journée Scientifique et Technique : Gestion des données et nouvel environnement numérique en géotechnique*. Paris, p. 10-11. URL : https://www.cfms-sols.org/documentation/exposes-du-cfms#2022 (cf. p. 217).

Bel, J. (2018). « Modélisation physique de l'impact du creusement d'un tunnel par tunnelier à front pressurisé sur des fondations profondes ». Thèse de doct. Université de Lyon (cf. p. 26, 28).

Bellmann, R. (1961). *Adaptive Control Processes : A Guided Tour*. Princeton University Press, Princeton (cf. p. 49).

Berthoz, N. (2012). « Modélisation physique et théorique du creusement pressurisé des tunnels en terrains meubles homogènes et stratifiés ». Thèse de doct. Laboratoire GéoMatériaux - Ecole Nationale des Travaux Publics de l'Etat (cf. p. 26).

Berthoz, N., D. Branque et D. Subrin (2020). « Déplacements induits par les tunneliers : rétro-analyse de chantiers en milieu urbain sur la base de calculs éléments finis en section courante ». *Revue Française de Géotechnique* 164, p. 1. DOI : 10.1051/geotech/2020019 (cf. p. 26, 27).

Bobbitt, Z. (2021). *Interpolation vs. Extrapolation : What's the Difference ?* URL : https://www.statology.org/interpolation-vs-extrapolation/ (cf. p. 79).

Bobet, A. (2001). « Analytical Solutions for Shallow Tunnels in Saturated Ground ». *Journal of Engineering Mechanics* 127.12, p. 1258-1266. DOI : 10.1061/(ASCE)0733-9399(2001)127:12(1258) (cf. p. 26).

Boser, B. E., I. M. Guyon et V. N. Vapnik (1992). « A Training Algorithm for Optimal Margin Classifiers ». *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. New York, NY, USA : Association for Computing Machinery, p. 144-152. DOI : 10.1145/130385.130401. URL : https://doi.org/10.1145/130385.130401 (cf. p. 51, 52).

Bouayad, D. et F. Emeriault (2017). « Modeling the relationship between ground surface settlements induced by shield tunneling and the operational and geological parameters based on the hybrid PCA/ANFIS method ». *Tunnelling and Underground Space Technology* 68.March, p. 142-152. DOI : 10.1016/j.tust.2017.03.011 (cf. p. 65, 70, 74).

Boubou, R., F. Emeriault et R. Kastner (2010). « Artificial neural network application for the prediction of ground surface movements induced by shield tunnelling ». *Canadian Geotechnical Journal* 47.11, p. 1214-1233. DOI : 10.1139/T10-023 (cf. p. 65, 67-69, 71, 74, 79, 81, 82).

Bouley, D. (2010). *Estimating a Data Center's Electrical Carbon Footprint*. Rapp. tech. APC by Schneider Electric, p. 1-13 (cf. p. 41).

Bourany, T. (2019). « Les 5V du big data ». *Regards croisés sur l'économie* n° 23.2, p. 27-31. DOI : 10.3917/rce.023.0027 (cf. p. 37).

Bourgeois, E., S. Burlon et F. Cuira (2018). « Modélisation numérique des ouvrages géotechniques ». *Technique de l'ingénieur* (cf. p. 27).

Breiman, L. (2001). « Random Forests ». *Machine Learning* 45, p. 5-32. DOI : 10.1023/A:1010933404324 (cf. p. 51, 55, 68).

Castelfranchi, C. (2013). « Alan Turing's "Computing Machinery and Intelligence" ». *Topoi* 32.2, p. 293-299. DOI : 10.1007/s11245-013-9182-y (cf. p. 34).

Chandola, V., A. Banerjee et V. Kumar (2009). « Anomaly detection : A survey ». *ACM Computing Surveys* 41.3, p. 1-58. DOI : 10.1145/1541880.1541882. URL : https://dl.acm.org/doi/10.1145/1541880.1541882 (cf. p. 50).

Charret, G., A. Arnaud, F. Berthoud, B. Bzeznik, A. Defize, Y. Delay, F. Drago, G. Feltin, N. Gibelin et G. Guennebaud (2022). *Estimation de l'empreinte carbone du stockage de données*. Rapp. tech. CNRS - GRICAD (cf. p. 41).

Chen, J., X. Shen et Q. Chen (2022). « Prediction of Maximum Surface Settlements of Bai-Hua Tunnel Section based on Machine Learning ». *Journal of Physics : Conference Series* 2185.1, p. 012042. DOI : 10.1088/1742-6596/2185/1/012042 (cf. p. 65, 74, 76).

Chen, R.-P., P. Zhang, X. Kang, Z.-Q. Zhong, Y. Liu et H.-N. Wu (2019a). « Prediction of maximum surface settlement caused by earth pressure balance (EPB) shield tunneling with ANN methods ». *Soils and Foundations* 59.2, p. 284-295. DOI : 10.1016/j.sandf.2018.11.005 (cf. p. 64, 65, 67, 71, 72, 74, 81, 115, 209, 242).

Chen, R., P. Zhang, H. Wu, Z. Wang et Z. Zhong (2019b). « Prediction of shield tunneling-induced ground settlement using machine learning techniques ». *Frontiers of Structural and Civil Engineering* 13.6, p. 1363-1378. DOI : 10.1007/s11709-019-0561-3 (cf. p. 64-66, 74).

Chen, T. et T. He (2014). « xgboost : Extreme Gradient Boosting ». *R Lecture* 2016, p. 1-84 (cf. p. 51, 55).

Chouinar, J.-C. (2022). *Decision Trees in Machine Learning, with Examples (Python)*. URL : https://www.jcchouinard.com/decision-trees-in-machine-learning/ (cf. p. 53).

Chu, X., I. F. Ilyas, S. Krishnan et J. Wang (2016). « Data cleaning : Overview and emerging challenges ». *Proceedings of the ACM SIGMOD International Conference on Management of Data* 26-June-20, p. 2201-2206. DOI : 10.1145/2882903.2912574 (cf. p. 38).

CNIL (2022). *Intelligence Artificielle, de quoi parle-t-on ?* URL : https://www.cnil.fr/fr/intelligence-artificielle/intelligence-artificielle-de-quoi-parle-t-on (cf. p. 36).

Cording, E. et W. Hansmire (1975). « Displacements around soft ground tunnels. General report : Session IV, tunnels in soil ». *5th Pan-American Congress on Soil Mechanics and Foundation Engineering*. Buenos Aires., p. 571-632 (cf. p. 25).

Cortes, C., V. Vapnik et L. Saitta (1995). « Support-vector networks ». *Machine Learning* 20.3, p. 273-297. DOI : 10.1007/BF00994018. URL : https://link.springer.com/article/10.1007/BF00994018 (cf. p. 51, 52).

CrowdFlower (2016). *Data Science Report*. Rapp. tech. URL : https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf (cf. p. 38).

Dargham, H. (2021). *Séance photo gare Arcueil Cachan Ligne 15 Sud-Ouest du Grand Paris Express* (cf. p. 18).

Dhiman, G. et V. Kumar (2019). « Seagull optimization algorithm : Theory and its applications for large-scale industrial engineering problems ». *Knowledge-Based Systems* 165, p. 169-196. DOI : 10.1016/j.knosys.2018.11.024. URL : https://doi.org/10.1016/j.knosys.2018.11.024 (cf. p. 82).

Dias, D. et R. Kastner (2013). « Movements caused by the excavation of tunnels using face pressurized shields — Analysis of monitoring and numerical modeling results ». *Engineering Geology* 152.1, p. 17-25. DOI : 10.1016/j.enggeo.2012.10.002 (cf. p. 27).

Dindarloo, S. R. et E. Siami-Irdemoosa (2015). « Maximum surface settlement based classification of shallow tunnels in soft ground ». *Tunnelling and Underground Space Technology* 49, p. 320-327. DOI : 10.1016/j.tust.2015.04.021 (cf. p. 65, 72, 74, 75).

Do, N. A. et D. Dias (2017). « A comparison of 2D and 3D numerical simulations of tunnelling in soft soils ». *Environmental Earth Sciences* 76.3, p. 1-12. DOI : 10.1007/s12665-017-6425-z (cf. p. 27).

Ebid, A. M. (2021). *35 Years of (AI) in Geotechnical Engineering : State of the Art*. T. 39. 2. Springer International Publishing, p. 637-690. DOI : 10.1007/s10706-020-01536-7 (cf. p. 62, 63).

Einstein, H. H. et C. W. Schwartz (1979). « Simplified Analysis for Tunnel Supports ». *Journal of the Geotechnical Engineering Division* 105.4, p. 499-518. DOI : 10.1061/AJGEB6.0000786 (cf. p. 26).

El Jirari, S. (2021). « Modélisation numérique du processus de creusement pressurisé des tunnels ». Thèse de doct. Université de Lyon (cf. p. 28).

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 3rd. O'Reilly Media, Inc., p. 861. URL : https://learning.oreilly.com/library/view/hands-on-machine-learning/9781098125967/ (cf. p. 47, 50, 73).

Gilleron, N. (2017). « Méthode de prévision des tassements provoqués par le creusement des tunnels urbains et influence des présoutènements » (cf. p. 27).

Gilleron, N. et E. Bourgeois (2018). « Modéliser une cuvette de tassement au tunnelier réaliste à l'aide d'une loi de comportement adaptée ». *Journées nationales de Géotechnique et de Géologie de l'ingénieur*. Champs-sur-Marne. URL : https://www.ifsttar.fr/collections/ActesInteractifs/AII3/pdfs/169587.pdf (cf. p. 27).

Gilleron, N., E. Bourgeois et A. Saïtta (2017). « Limites de la modélisation bidimensionnelle des tunnels urbains pour la prévision des tassements ». *Revue Française de Géotechnique* 150, p. 2. DOI : 10.1051/geotech/2017003 (cf. p. 29).

Goh, A. T. et A. M. Hefney (2010). « Reliability assessment of EPB tunnel-related settlement ». *Geomechanics and Engineering* 2.1, p. 57-69. DOI : 10.12989/gae.2010.2.1.057 (cf. p. 65, 74).

Guglielmetti, V., P. Grasso, A. Mahtab et S. Xu (2008). *Mechanized Tunnelling in Urban Areas*. Taylor & Francis Group (cf. p. 22).

Guilloux, A. (2016). « Les projets d'ouvrages géotechniques : apports de l'observation et de la modélisation ». *Revue Française de Géotechnique* 146, p. 1. DOI : 10.1051/geotech/2016001 (cf. p. 29).

Guilloux, A., H. Le Bissonnais, M. Cahn et J. P. Janin (2021). « Creusement des tunnels - Méthodes de construction et géotechnique ». *Travaux publics et infrastructures* 33.0. DOI : 10.51257/a-v1-c5583 (cf. p. 10, 22).

Hajihassani, M., R. Kalatehjari, A. Marto, H. Mohamad et M. Khosrotash (2020). « 3D prediction of tunneling-induced ground movements based on a hybrid ANN and empirical methods ». *Engineering with Computers* 36.1, p. 251-269. DOI : 10.1007/s00366-018-00699-5 (cf. p. 64, 65, 74, 82).

Hasanipanah, M., M. Noorian-Bidgoli, D. Jahed Armaghani et H. Khamesi (2016). « Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling ». *Engineering with Computers* 32.4, p. 705-715. DOI : 10.1007/s00366-016-0447-0 (cf. p. 64, 65, 67, 74, 82).

Herrenknecht (2022). *EPB Shield*. URL : https://www.herrenknecht.com/en/products/productdetail/epb-shield/ (cf. p. 12, 13).

Herrenknecht, M., M. Thewes et C. Budach (2011). « The development of earth pressure shields : from the beginning to the present ». *Geomechanics und Tunnelling* 4.1, p. 11-35. DOI : 10.1002/geot.201100003 (cf. p. 11).

Huat, C. Y., D. J. Armaghani, E. Momeni et S. H. Lai (2023). « Empirical, Statistical, and Machine Learning Techniques for Predicting Surface Settlement Induced by Tunnelling ». *Artificial Intelligence in Mechatronics and Civil Engineering : Bridging the Gap*. Sous la dir. d'E. Momeni, D. Jahed Armaghani et A. Azizi. Singapore : Springer Nature Singapore, p. 39-77. DOI : 10.1007/978-981-19-8790-8{_}2 (cf. p. 63).

IBM (2016). *A glimpse inside the mind of a data scientist*. Rapp. tech. URL : https://www.ibm.com/downloads/cas/W6GEX9LL (cf. p. 38).

Janin, J.-P. (2017). « Apports de la simulation numérique tridimensionnelle dans les études de tunnels ». *Revue Française de Géotechnique* 150, p. 3. DOI : 10.1051/geotech/2017006 (cf. p. 27).

Jong, S., D. Ong et E. Oh (2021). « State-of-the-art review of geotechnical-driven artificial intelligence techniques in underground soil-structure interaction ». *Tunnelling and Underground Space Technology* 113.March, p. 103946. DOI : 10.1016/j.tust.2021.103946 (cf. p. 62).

Karakus, M. (2007). « Appraising the methods accounting for 3D tunnelling effects in 2D plane strain FE analysis ». *Tunnelling and Underground Space Technology* 22.1, p. 47-56. DOI : 10.1016/j.tust.2006.01.004 (cf. p. 27).

Kasper, T. et G. Meschke (2004). « A 3D finite element simulation model for TBM tunnelling in soft ground ». *International Journal for Numerical and Analytical Methods in Geomechanics* 28.14, p. 1441-1460. DOI : 10.1002/nag.395 (cf. p. 28).

Keerthana, V. (2022). *What, why and how of Spectral Clustering !* URL : https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/ (cf. p. 49).

Kim, C. Y., G. J. Bae, S. W. Hong, C. H. Park, H. K. Moon et H. S. Shin (2001). « Neural network based prediction of ground surface settelements due to tunnelling ». *Computers and Geotechnics* 28.6-7, p. 517-547. DOI : 10.1016/S0266-352X(01)00011-8 (cf. p. 65, 74).

Kim, D., K. Kwon, K. Pham, J. Y. Oh et H. Choi (2022a). « Surface settlement prediction for urban tunneling using machine learning algorithms with Bayesian optimization ». *Automation in Construction* 140.January, p. 104331. DOI : 10.1016/j.autcon.2022.104331 (cf. p. 65, 66, 74, 76, 82).

Kim, D., K. Pham, J.-Y. Oh, S.-J. Lee et H. Choi (2022b). « Classification of surface settlement levels induced by TBM driving in urban areas using random forest with data-driven feature selection ». *Automation in Construction* 135.August 2021, p. 104109. DOI : 10.1016/j.autcon.2021.104109 (cf. p. 65, 66, 68, 70, 71, 74, 75, 82).

Klein, A. et W. Lehner (2009). « Representing data quality in sensor data streaming environments ». *Journal of Data and Information Quality* 1.2. DOI : 10.1145/1577840.1577845 (cf. p. 38, 216).

Kohestani, V. R., M. Bazargan-Lari et J. Asgari-marnani (2017). « Prediction of maximum surface settlement caused by earth pressure balance shield tunneling using random forest ». *Journal of AI and Data Mining* 5.1, p. 127-135. DOI : 10.22044/JADM.2016.748 (cf. p. 65, 71, 72, 74).

Krishnan, S., D. Haas, M. J. Franklin et E. Wu (2016). « Towards reliable interactive data cleaning : A user survey and recommendations ». *HILDA 2016 - Proceedings of the Workshop on Human-In-the-Loop Data Analytics* 1, p. 1-5. DOI : 10.1145/2939502.2939511 (cf. p. 38, 216).

Larousse (2023a). *Prédire*. URL : https://www.larousse.fr/dictionnaires/francais/pr%C3%A9dire/63424 (cf. p. 147).

Larousse (2023b). *Prévoir*. URL : https://www.larousse.fr/dictionnaires/francais/pr%C3%A9voir/63883 (cf. p. 147).

Laurae (2016). *xgboost : "Hi I'm Gamma. What can I do for you ?" — and the tuning of regularization*. URL : https://medium.com/data-design/xgboost-hi-im-gamma-what-can-i-do-for-you-and-the-tuning-of-regularization-a42ea17e6ab6 (cf. p. 163).

Lebdaoui, S. (2022). *Apport de l'intelligence artificielle dans la prédiction des tassements causés par le passage du tunnelier*. Rapp. tech. Paris : Setec terrasol ; ENSEIRB MATMECA (cf. p. 152, 209).

LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard et L. Jackel (1989). « Handwritten Digit Recognition with a Back-Propagation Network ». *Advances in Neural Information Processing Systems*. Sous la dir. de D. Touretzky. T. 2. Morgan-Kaufmann. URL : https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf (cf. p. 51).

Leroux, V. et A. Campeau (2018). « Tunnel Database : An Information System Useful for Underground Construction in Montreal ». *71st Canadian Geotechnical Conference*, p. 2-9. URL : https://members.cgs.ca/conferences/GeoEdmonton/papers/geo2018Paper616.pdf (cf. p. 66).

Likitlersuang, S., C. Surarak, S. Suwansawat, D. Wanatowski, E. Oh et A. Balasubramaniam (2014). « Simplified finite-element modelling for tunnelling-induced settlements ». *Geotechnical Research* 1.4, p. 133-152. DOI : 10.1680/gr.14.00016 (cf. p. 27).

Liu, L., W. Zhou et M. Gutierrez (2022). « Effectiveness of predicting tunneling-induced ground settlements using machine learning methods with small datasets ». *Journal of Rock Mechanics and Geotechnical Engineering* 14.4, p. 1028-1041. DOI : 10.1016/j.jrmge.2021.08.018 (cf. p. 64-66, 68, 71, 72, 74-77, 80, 176).

Loganathan, N. et H. G. Poulos (1998). « Analytical Prediction for Tunneling-Induced Ground Movements in Clays ». *Journal of Geotechnical and Geoenvironmental Engineering* 124.9, p. 846-856. DOI : 10.1061/(asce)1090-0241(1998)124:9(846) (cf. p. 26).

Mahmoodzadeh, A., M. Mohammadi, A. Daraei, H. Farid Hama Ali, N. Kameran Al-Salihi et R. Mohammed Dler Omer (2020). « Forecasting maximum surface settlement caused by urban tunneling ». *Automation in Construction* 120.August, p. 103375. DOI : 10.1016/j.autcon.2020.103375 (cf. p. 65, 74).

Mair, R. J. et R. N. Taylor (1997). « Bored tunnelling in the urban environment ». *Soil mechanics and foundation engineering* 14, p. 2353-2386. URL : https://www.issmge.org/uploads/publications/1/31/1997_04_0049.pdf (cf. p. 19, 22, 23).

Marinos, V., G. Prountzopoulos, P. Fortsakis, D. Koumoutsakos, K. Korkaris et D. Papouli (2013). « "Tunnel Information and Analysis System" : A Geotechnical Database for Tunnels ». *Geotechnical and Geological Engineering* 31.3, p. 891-910. DOI : 10.1007/s10706-012-9570-x. URL : http://link.springer.com/10.1007/s10706-012-9570-x (cf. p. 66, 108).

Marto, A., M. Hajihassani, R. Kalatehjari, E. Namazi et H. Sohaei (2012). « Simulation of longitudinal surface settlement due to tunnelling using artificial neural network ». *International Review on Modelling and Simulations* 5.2, p. 1024-1031 (cf. p. 64, 65, 67, 74).

Migliazza, M., M. Chiorboli et G. P. Giani (2009). « Comparison of analytical method, 3D finite element model with experimental subsidence measurements resulting from the extension of the Milan underground ». *Computers and Geotechnics* 36.1-2, p. 113-124. DOI : 10.1016/j.compgeo.2008.03.005. URL : http://dx.doi.org/10.1016/j.compgeo.2008.03.005 (cf. p. 28).

Moeinossadat, S. R., K. Ahangari et K. Shahriar (2018). « Modeling maximum surface settlement due to EPBM tunneling by various soft computing techniques ». *Innovative Infrastructure Solutions* 3.1, p. 1-13. DOI : 10.1007/s41062-017-0114-3 (cf. p. 65, 74).

Mohammadi, S. D., F. Naseri et S. Alipoor (2015). « Development of artificial neural networks and multiple regression models for the NATM tunnelling-induced settlement in Niayesh subway tunnel, Tehran ». *Bulletin of Engineering Geology and the Environment* 74.3, p. 827-843. DOI : 10.1007/s10064-014-0660-2 (cf. p. 65, 67, 71, 74).

Moller, S. et P. Vermeer (2008). « On numerical simulation of tunnel installation ». *Tunnelling and Underground Space Technology* 23.4, p. 461-475. DOI : 10.1016/j.tust.2007.08.004 (cf. p. 27, 29).

Moller, S. (2006). *Tunnel Induced Settlements and Structural Forces in Linings*, p. 174. URL : https://www.igs.uni-stuttgart.de/dokumente/Mitteilungen/54_Moeller.pdf (cf. p. 19, 20).

Moor, J. (2006). « The Dartmouth College Artificial Intelligence Conference : The next fifty years ». *AI Magazine* 27.4, p. 87-91 (cf. p. 34).

Mroueh, H. et I. Shahrour (2008). « A simplified 3D model for tunnel construction using tunnel boring machines ». *Tunnelling and Underground Space Technology* 23.1, p. 38-45. DOI : 10.1016/j.tust.2006.11.008 (cf. p. 28).

Mullarkey, P. W. et S. J. Fenves (1986). « Fuzzy logic in a geotechnical knowledge-based system :CONE ». *Civil Engineering Systems* 3.2, p. 58-81. DOI : 10.1080/02630258608970429 (cf. p. 62).

Nallaperumal (2021). *Calculation of Bias & Variance in python*. URL : https://medium.com/analytics-vidhya/calculation-of-bias-variance-in-python-8f96463c8942 (cf. p. 43).

Neaupane, K. M. et N. R. Adhikari (2006). « Prediction of tunneling-induced ground movement with the multi-layer perceptron ». *Tunnelling and Underground Space Technology* 21.2, p. 151-159. DOI : 10.1016/j.tust.2005.07.001 (cf. p. 25, 65, 67, 72, 74).

O'Reilly, M. et B. New (1982). « Settlements above tunnels in the United Kingdom - their magnitude and prediction ». *Tunnelling*. London : Institution of Mining et Metallurgy, p. 173-181 (cf. p. 22, 25).

Ocak, I. et S. E. Seker (2013). « Calculation of surface settlements caused by EPBM tunneling using artificial neural network, SVM, and Gaussian processes ». *Environmental Earth Sciences* 70.3, p. 1263-1276. DOI : 10.1007/s12665-012-2214-x (cf. p. 65, 68, 70-72, 74).

Oracle (2022). *Qu'est-ce qu'une base de données ?* URL : https://www.oracle.com/ca-fr/database/what-is-database/ (cf. p. 40).

Padhma, M. (2021). *Loss functions to evaluate Regression Models*. URL : https://medium.com/analytics-vidhya/loss-functions-to-evaluate-regression-models-8dac47e327e2 (cf. p. 56).

Panet, M. et A. Guenot (1982). « Analysis of convergence behind the face of a tunnel », p. 197-204. DOI : 10.1016/0148-9062(83)91744-8 (cf. p. 27).

Panet, M. (1995). *Le Calcul des Tunnels par la Methode Convergence - Confinement*. Sous la dir. de Presses de l'école nationale des Ponts et Chaussées (cf. p. 27).

Panet, M. et J. Sulem (2021). *Le calcul des tunnels par la méthode convergence-confinement* (cf. p. 27).

Peck, R. B. (1969). « Deep excavations and tunneling in soft ground (State of the art report) ». *7th International Conference on Soil Mechanics and Foundation Engineering*. Mexico, p. 225-290. URL : https://www.issmge.org/publications/publication/deep-excavations-and-tunneling-in-soft-ground (cf. p. 19, 21, 22, 25).

PlanèteTP (2022). *L'histoire des tunnels*. URL : http://www.planete-tp.com/les-tunnels-r203.html (cf. p. 10).

PostgreSQLTutorial (2023). *PostgreSQL Joins* (cf. p. 259).

Pourtaghi, A. et M. A. Lotfollahi-Yaghin (2012). « Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling ». *Tunnelling and Underground Space Technology* 28.1, p. 257-271. DOI : 10.1016/j.tust.2011.11.008 (cf. p. 65, 71, 72, 74).

Premanand, S. (2021). *The A-Z guide to Support Vector Machine*. URL : https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/ (cf. p. 53).

Qiao, J., J. Liu, W. Guo et Y. Zhang (2010). « Artificial neural network to predict the surface maximum settlement by shield tunneling ». *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6424 LNAI.PART 1, p. 257-265. DOI : 10.1007/978-3-642-16584-9{_}24 (cf. p. 65, 74).

Quantmetry (2021). *IA de confiance - du concept à l'action*. Rapp. tech. URL : https://www.quantmetry.com/lp-ia-de-confiance/ (cf. p. 35).

Quantmetry (2023). *La future réglementation de l'UE sur l'Intelligence Artificielle*. Rapp. tech., p. 1. URL : https://youtu.be/EDnB_TfXDek?t=968 (cf. p. 35).

Quinlan, J. R. (1986). « Induction of decision trees ». *Machine Learning* 1.1, p. 81-106. DOI : 10.1007/BF00116251. URL : https://link.springer.com/article/10.1007/BF00116251 (cf. p. 51, 52).

Rakotomalala, R. (2023). *Régression régularisée. Ridge-Lasso-Elasticnet*. Rapp. tech. URL : https://eric.univ-lyon2.fr/ricco/cours/slides/regularized_regression.pdf (cf. p. 44).

Richa, T., S. Lebdaoui, J.-M. Pereira, G. Chapron, F. Lanquette et L.-M. Guayacán-Carrillo (2023). « De l 'apprentissage automatique pour prévoir les tassements induits par le creusement au tunnelier – Application à 2 lignes du Grand Paris Express ». *AFTES*. Paris, p. 1-8 (cf. p. 96).

Richa, T., J.-M. Pereira, G. Chapron et L. M. Guayacán-Carrillo (2022). « Constitution d'une base de données des mesures obtenues lors du creusement de deux tunnels du Grand Paris Express ». *JNGG* (cf. p. 97, 101).

Sagaseta, C. (1987). « Analysis of undrained soil deformation due to ground loss ». *Géotechnique* 37.3, p. 301-320. DOI : 10.1680/geot.1987.37.3.301 (cf. p. 26).

Salperwyck, C. (2013). « Apprentissage incrémental en-ligne sur flux de données ». Thèse de doct. Université Charles de Gaulle - Lille III (cf. p. 47, 48).

Santos, O. J. et T. B. Celestino (2008). « Artificial neural networks analysis of São Paulo subway tunnel settlement data ». *Tunnelling and Underground Space Technology* 23.5, p. 481-491. DOI : 10.1016/j.tust.2007.07.002 (cf. p. 65, 71-74).

Sckit-learn (2011). *Cross-validation : evaluating estimator performance*. URL : https://scikit-learn.org/stable/modules/cross_validation.html (cf. p. 46).

Sckit-learn (2023). *Ensemble methods*. URL : https://scikit-learn.org/stable/modules/ensemble.html#random-forest-parameters (cf. p. 161).

Shi, J., J. A. R. Ortigao et J. Bai (1998). « Modular Neural Networks for Predicting Settlements During Tunneling ». *Journal of Geotechnical and Geoenvironmental Engineering* 124.May, p. 389-395 (cf. p. 61, 63, 65, 74).

SociétéGrandParis (2021). *La réalisation des tunnels*. URL : https://youtu.be/yH2aGaBpByM?list=PLrTVdloHpSdVkeY-CDmzGbzudJV6Ijwv_ (cf. p. 15).

SociétéGrandParis (2022). *L'essentiel du Grand Paris Express*. URL : https://www.societedugrandparis.fr/nouveau-metro/grand-paris-express (cf. p. 90).

Su, J., Y. Wang, X. Niu, S. Sha et J. Yu (2022). « Prediction of ground surface settlement by shield tunneling using XGBoost and Bayesian Optimization ». *Engineering Applications of Artificial Intelligence* 114.June, p. 105020. DOI : 10.1016/j.engappai.2022.105020 (cf. p. 65, 68, 71, 74, 76, 82).

Sumathi, S. et S. Esakkirajan (2007). *Fundamentals of Relational Database Management Systems*. T. 47. Studies in Computational Intelligence. Berlin, Heidelberg : Springer Berlin Heidelberg. DOI : 10.1007/978-3-540-48399-1. URL : http://link.springer.com/10.1007/978-3-540-48399-1 (cf. p. 40).

Suwansawat, S. (2002). « Earth pressure balance (EPB) shield tunneling in Bangkok : ground response and prediction of surface settlements using artificial neural networks ». Thèse de doct. Massachusetts Institute of Technology, Cambridge, MA, USA, p. 597 (cf. p. 20).

Suwansawat, S. et H. H. Einstein (2006). « Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling ». *Tunnelling and Underground Space Technology* 21.2, p. 133-150. DOI : 10.1016/j.tust.2005.06.007 (cf. p. 65, 67, 70-72, 74, 80, 81).

Talend (2022a). *Structured vs. Unstructured Data : A Complete Guide*. URL : https://www.talend.com/resources/structured-vs-unstructured-data/ (cf. p. 216).

Talend (2022b). *What are Data Silos ?* URL : https://www.talend.com/resources/what-are-data-silos/ (cf. p. 216).

Talend (2023). *Tout savoir sur le traitement des données*. URL : https://www.talend.com/fr/resources/what-is-data-processing/ (cf. p. 95).

Taylor, R. N. (1995). *Geotechnical Centrifuge Technology*. DOI : 10.4324/9780203210536 (cf. p. 28).

Terrasol (2023). *SOCATOP tronçons VL1 et VL2 - Terrasol*. URL : https://www.terrasol.fr/realisations/socatop-tron%C3%A7ons-vl1-et-vl2 (cf. p. 10).

Trevisan, V. (2022). *Comparing Robustness of MAE, MSE and RMSE*. URL : https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828 (cf. p. 57).

Turing, A. M. (1950). « Computing Machinery and Intelligence ». *Mind* 49, p. 433-460. URL : https://www.csee.umbc.edu/courses/471/papers/turing.pdf (cf. p. 34, 36).

U/ErockLobster (2019). *SQL Join Chart - Custom Poster Size*. URL : https://www.reddit.com/r/SQL/comments/aysflk/sql_join_chart_custom_poster_size/ (cf. p. 259).

Verruijt, A. et J. R. Booker (1998). « Surface settlements due to deformation of a tunnel in an elastic half plane ». *Géotechnique* 48.5, p. 709-713. DOI : 10.1680/geot.1998.48.5.709 (cf. p. 26).

Vinci (2023). *Duplex A86 - VINCI Construction Grands Projets*. URL : https://www.vinci-construction-projets.com/fr/realisations/duplex-a86/ (cf. p. 10).

Waldmann, R. (2005). « L'histoire des tunnels ». *Tunnels et ouvrages souterrains* 188, p. 81-88 (cf. p. 10).

Wang, F., B. Gou et Y. Qin (2013). « Modeling tunneling-induced ground surface settlement development using a wavelet smooth relevance vector machine ». *Computers and Geotechnics* 54, p. 125-132. DOI : 10.1016/j.compgeo.2013.07.004 (cf. p. 65, 67, 71, 72, 74, 79, 83).

Wickham, H. (2014). « Tiday Data ». *Journal of Statistical Software* 59.10, p. 1-23. URL : http://www.jstatsoft.org/ (cf. p. 103).

Withers, A., S. Pontin et M. Black (2000). « The geotechnical database for the Jubilee Line Extension project ». *Proceedings of the Institution of Civil Engineers - Geotechnical Engineering* 143.3, p. 131-138. DOI : 10.1680/geng.2000.143.3.131 (cf. p. 66).

Ye, A. (2020). *Real Artificial Intelligence : Understanding Extrapolation vs Generalization*. URL : https://towardsdatascience.com/real-artificial-intelligence-understanding-extrapolation-vs-generalization-b8e8dcf5fd4b (cf. p. 78).

Zhang, P., H. N. Wu, R. P. Chen et T. H. Chan (2020a). « Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction : A comparative study ». *Tunnelling and Underground Space Technology* 99.May 2019, p. 103383. DOI : 10.1016/j.tust.2020.103383 (cf. p. 64, 65, 68, 71, 72, 74, 76, 82).

Zhang, W., H. R. Li, C. Z. Wu, Y. Q. Li, Z. Q. Liu et H. L. Liu (2021a). « Soft computing approach for prediction of surface settlement induced by earth pressure balance shield tunneling ». *Underground Space (China)* 6.4, p. 353-363. DOI : 10.1016/J.UNDSP.2019.12.003 (cf. p. 65, 68, 71, 74).

Zhang, W., H. Li, Y. Li, H. Liu, Y. Chen et X. Ding (2021b). « Application of deep learning algorithms in geotechnical engineering : a short critical review ». *Artificial Intelligence Review* 54.8, p. 5633-5673. DOI : 10.1007/s10462-021-09967-1 (cf. p. 62).

Zhang, W., R. Zhang, C. Wu, A. T. C. Goh, S. Lacasse, Z. Liu et H. Liu (2020b). « State-of-the-art review of soft computing applications in underground excavations ». *Geoscience Frontiers* 11.4, p. 1095-1106. DOI : 10.1016/j.gsf.2019.12.003. URL : https://doi.org/10.1016/j.gsf.2019.12.003 (cf. p. 62).

Zhou, J., X. Shi, K. Du, X. Qiu, X. Li et H. S. Mitri (2017). « Feasibility of Random-Forest Approach for Prediction of Ground Settlements Induced by the Construction of a Shield-Driven Tunnel ». *International Journal of Geomechanics* 17.6. DOI : 10.1061/(asce)gm.1943-5622.0000817 (cf. p. 25).

Zhou, X., C. Zhao et X. Bian (2023). « Prediction of maximum ground surface settlement induced by shield tunneling using XGBoost algorithm with golden-sine seagull optimization ». *Computers and Geotechnics* 154.July 2022, p. 105156. DOI : 10.1016/j.compgeo.2022.105156 (cf. p. 65, 67, 71, 74, 82).

Liste des Figures

Liste des Tables

Liste des Scripts

Liste des Acronymes

Terme	Description
TBM	Tunnel Boring Machine = Tunnelier
EPB	Earth Pressure Balance = Tunnelier à Pression de Terre
$V_{tunnelier}$ [mm/min]	Vitesse d'avancement du tunnelier
G_H [mm]	Guidage horizontal du tunnelier
G_V [mm]	Guidage vertical du tunnelier
P_{totale} [kN]	Poussée totale du tunnelier
$P_{molettes}$ [kN]	Poussée exercée sur les molettes
P_{front} [bar]	Pression au front du tunnel
$\tau_{remplissage}$ [%]	Taux de remplissage de terre de la chambre d'abattage
V_{eau} [m^3]	Volume d'eau injecté dans la chambre d'abattage
$V_{polymères}$ [m^3]	Volume de polymères injecté dans la chambre d'abattage
RDC	Roue de Coupe du tunnelier
V_{RDC} [tour/min]	Vitesse de rotation de la roue de coupe
M_{RDC} [kN.m]	Moment de rotation (couple) de la roue de coupe
L_{RDC} [mm]	Pénétration de la roue coupe
P_{RDC} [kW]	Puissance de la roue de coupe
E_{RDC} [mJ/m^3]	Energie de la roue de coupe
V_{vis} [tour/min]	Vitesse de rotation de la vis d'extraction
M_{vis} [kN.m]	Moment de rotation de la vis d'extraction
P_{vis} [bar]	Pression des terres au milieu de la vis d'extraction
$R_{vis-avancement}$	Ratio de la vitesse de la vis sur la vitesse d'avancement du tunnelier
P_{peson1} [ton]	Poids brut de déblais mesuré par le 1^{er} peson
P_{peson2} [ton]	Poids brut de déblais mesuré par le 2^{nd} peson
$P_{vérins}$ [bar]	Pression de poussée d'un groupe de vérins
$P_{mortier}$ [bar]	Pression moyenne du mortier de bourrage injecté à l'arrière de la jupe
$V_{mortier}$ [m^3]	Volume Total de mortier de bourrage injecté à l'arrière de la jupe
$V_{mortier\ théorique}$ [m^3]	
$N_{coups\ mortier}$	Nombre de coups d'injection de mortier de bourrage à l'arrière de la jupe
PK	position absolue sur le linéaire d'un tunnel, à partir d'un point de référence souvent variable dans la vie d'un projet, et exprimé en kilomètres

Terme	Description
PM	position absolue sur le linéaire d'un tunnel, à partir d'un point de référence souvent variable dans la vie d'un projet, et exprimé en mètres
pk_{rdc}	PK de la roue de coupe du tunnelier
DDE	Date de Début d'Excavation d'un anneau
DFE	Date de Fin d'Excavation d'un anneau
DDP	Date de Début de Pose d'un anneau
DFP	Date de Fin de Pose d'un anneau
p_{nappe} [m]	Profondeur du tunnel par rapport à la nappe
MSPT	Modified Standard Penetration Test (Chen et al., 2019a)
MDPT	Modified Dynamic Penetration Test (Chen et al., 2019a)
MUCS	Modified Uniaxial Compressive Strength (Chen et al., 2019a)
s_{trans}	Tassement transversal
s_{long}	Tassement longitudinal pour une position fixe du front du tunnel
s_{3D}	Tassement tridimensionnel pour une position fixe du front du tunnel
x	Coordonnée le long de l'axe du tunnel
x_f	Position du front du tunnelier à un instant t
d_{front}	Distance au front de taille du tunnel
y	Coordonnée dans la direction orthogonale à l'axe du tunnel
d_{axe}	Distance horizontale à l'axe du tunnel
z	Coordonnée de la profondeur
s_{max}	Tassement maximal observé à l'axe du tunnel
s^*	Tassement maximal observé à une certaine distance à l'axe du tunnel
η	Ratio du tassement au front du tunnel par rapport au tassement maximal s*
$d_{front\ s*}$	Distance au front du tunnel pour laquelle le tassement atteint 95% de sa valeur maximale s*
s	Tassement induit en surface par le creusement du tunnel
s_0	Valeur de remise à 0 du tassement
i_y	Distance horizontale entre l'axe du tunnel et le point d'inflexion de la courbe de tassement transversal
i_x	facteur de forme qui rend compte de la position du point d'inflexion de la dérivée de la gaussienne cumulée

Terme	Description
$G_{[\mu,\sigma]}(\alpha)$	Fonction de répartition d'une loi Normale de moyenne μ et d'écart-type σ
m_x	coefficient pour décaler la courbe de tassement longitudinal
m_y	coefficient pour décaler la courbe de tassement transversal
k	Constante adimensionnelle, dépend du type de sol et informe sur la largeur de la cuvette de tassement transversale
V_s	Volume de la cuvette transversale de tassement
D	Diamètre d'excavation du tunnel
H [m]	Couverture du tunnel calculée entre le terrain naturel et l'axe du tunnel
C [m]	Couverture du tunnel calculée entre le terrain naturel et la voûte du tunnel
z_{TN} [m]	Côte du Terrain Naturel
ZIG	Zone d'Influence Géotechnique
h	Distance entre le terrain naturel et la base de la couche de sol
e	Épaisseur de la couche de sol
α	Coefficient rhéologique
E_M [MPa]	Module pressiométrique de Ménard
c [kPa]	Cohésion
φ [°]	Angle de frottement
γ [kN/m^3]	Poids volumique
K_0	Coefficient de pression des terres au repos
CGi	Calcaire grossier inférieur
CGm	Calcaire grossier moyen
AV	Argiles Vertes
AP	Argile Plastique
MA	Marnes Supragypseuses: Marnes d'Argenteuil
MP	Marnes Supragypseuses: Marnes de Pantin
MG	Masses et Marnes du Gypse
MM	Marnes de Meudon
IA	Intelligence Artificielle
SGBD	Système de Gestion de Base de Données = DataBase Management System (DBMS)
BD	Base de Données = Database
SQL	Structured Query Language

Terme	Description
DER	Diagramme Entité-Relation = Entity Relationship Diagram
EDA	Exploratory Data Analysis = Analyse Exploratoire des Données
OCR	Optical Character Recognition = reconnaissance optique de caractères
ML	Machine Learning = apprentissage automatique ou, par abus de langage, apprentissage machine
CV	Cross-Validation = validation croisée
k-NN	k Nearest Neighbors = les k plus proches voisins
LR	Linear Regression = Régression Linéaire
MLP	Multiple Linear Regression
MARS	Multivariate Adaptive Regression Spline
IF	Isolation Forest = Forêt d'Isolation
ANN	Artificial Neural Network = réseaux de neurones artificiels
BPNN	Back-Propagation Neural Network
GRNN	General Regression Neural Network
RBFNN	Radial Basis Function Neural Network
WNN	Wavelet Neural Network
EN	Elastic Net
DL	Deep Learning = apprentissage profond
LSTM	Long Short-Term Memory networks = Réseau récurrent à longue mémoire court terme
SVM	Support Vector Machine = Séparateur à Vaste Marge
DT	Decision Tree = arbre de décision
RF	Random Forest = forêt aléatoire
ELM	Extreme Learning Machine
GBM	Gradient Boosting Machine
LGBM	Light Gradient Boosting Machine
XGBoost	Extreme Gradient Boosting Machine
ANFIS	Adaptive Neuro-Fuzzy Inference System
GP	Gaussian Processes
GEP	Gene Expression Programming
NGS	Neuro-Genetic System
PCA	Principal Component Analysis = analyse en composantes principales
PSO	Particle Swarm Optimization
wsRVM	Smooth Relevance Vector Machine with a wavelet kernel
FI	Feature Importance = importance des caractéristiques

Terme	Description
FPI	Feature Permutation Importance
R	Coefficient de corrélation de Pearson
R^2	Coefficient de détermination = coefficient of determination
MAE	Mean Absolute Error = erreur absolue moyenne
QE	Quantile Error = erreur quantile
ME	Mean Error = erreur moyenne
RAE	Relative Absolute Error = erreur relative absolue
MSE	Mean Squared Error = erreur quadratique moyenne
RMSE	Root Mean Square Error = racine carrée de l'erreur quadratique moyenne
RRMSE	Relative Root Mean Square Error = racine carrée de l'erreur quadratique moyenne relative
MAPE	Mean Absolute Percentage Error = erreur relative absolue moyenne
MRE	Mean Absolute Percentage Error = erreur relative absolue moyenne
SGP	Société du Grand Paris, maître d'ouvrage du projet du Grand Paris Express
GPE	Projet du Grand Paris Express
L14S	Ligne 14 Sud du projet du Grand Paris Express
GC02	Tronçon GC02 de la ligne 14 Sud du Grand Paris Express
L14S2	Ligne 14 Sud du projet du Grand Paris Express, tronçon GC02
L15SO	Ligne 15 Sud-Ouest du projet du Grand Paris Express
TR1	Tronçon central de la ligne 15 Sud-Ouest
TR2	Tronçon est de la ligne 15 Sud-Ouest
TR3	Tronçon est de la ligne 15 Sud-Ouest
EF	Méthode des éléments finis
EF 2D	Méthode des éléments finis bidimensionnels
EF 3D	Méthode des éléments finis tridimensionnels
CV-CF	Methode convergence-confinement
λ	Taux de déconfinement
NATM	New Australian Tunelling Method

Annexes

Code d'extraction des données de mesures de tassements

A

Librairies →

```
Entrée [32]: import requests
             import json
             import pandas as pd
             from ws_tools import write_to_pretty_json
             import datetime as dt
```

Données à extraire →

```
Entrée [33]: LIST_COLS = [
                 'id',
                 'X',
                 'Y',
                 'dateOfValue',
                 'isSymbolCustom',
                 'last_alarm_level',
                 'last_value',
                 'metaid',
                 'move_decimals',
                 'move_unit',
                 'name',
                 'offsetX',
                 'offsetY',
                 'provider',
             ]
```

Lien WEB →

```
Entrée [34]: HEADERS = {
                 'Host': 'l1st3c.s
                 'Connection': 'ke
                 'Content-Length':
                 'Accept': 'app
                 'User-Agent':
                 'Content-type'
                 'Origin': 'htt
                 'Referer': 'ht
                 'Accept-Encoding':
                 'Accept-Language':
             #     'Cookie': 'SESSI                                    }',
             }
```

```
Entrée [35]: BASELINK = 'http://                    .com/'
```

Liste sections

```
Entrée [37]: DICT_SECTIONS = {
                 'TR1A': '{                                          }',
                 'TR1B': '{                                          }',
                 'TR1C': '{                                          }',
                 'TR1D': '{                                          }',
                 'TR1E': '{                                          }',
                 'TR1F': '{                                          }',
                 'TR1G': '{                                          }',
                 'TR1H': '{                                          }',
                 'TR2A': '{                                          }',
                 'TR2B': '{                                          }',
                 'TR2C': '{                                          }',
                 'TR2D': '{                                          }',
                 'TR2E': '{                                          }',
                 'TR2F': '{                                          }',
                 'TR2G': '{                                          }',
                 'TR3A': '{                                          }',
                 'TR3B': '{                                          }'
             }
```

→ Lien pour chacune des sections

Liste layers

Fonction pour extraire les layers

```
Entrée [38]: def get_layer_list(section):

    payload = {
        "Format":"GCFMessage",
        "Version":"0.1",
        "Timestamp":"2020-11-03T15:35:52.178Z",
        "Type":262,
        "GUID":"b65f2a19-f2d2-4095-9a5a-3bfb374951b1",
        "MessageID":4,
        "ObjectName":"WebSessionService.ProjectHelper",
        "MethodName":"getGisInfos",
        "Args":[
            "L15T3C",
            DICT_SECTIONS[section],
            "f361b438879287bf768a1d320967fe88"
        ],
        "SessionID":-1
    }

    i = 0
    while i < 10:
        res = requests.post(BASELINK, data=json.dumps(payload), headers=HEADERS).json()
        if 'result' in res['Result']:
            break
        else:
            i += 1

    if i == 10:
            raise Exception("Problème d'extraction get_layer_list")

    return [
        l['sublayers'][0]['name']
        for l in res['Result']['result']['layers']
        if l['sublayers'][0]['name'].startswith('Ent') and l['name'] != 'Inclino'
    ]
```

Liste entities

Fonction pour extraire les caractéristiques des capteurs

```
Entrée [39]: def get_entity_table(section, list_entnames):

    dict_entities = {c: [] for c in LIST_COLS + ['Entities']}
    for entname in list_entnames:
        payload = {
            "Format": "GCFMessage",
            "Version": "0.1",
            "Timestamp": "2020-11-03T14:57:29.931Z",
            "Type": 262,
            "GUID": "b65f2a19-f2d2-4095-9a5a-3bfb374951b1",
            "MessageID": 4,
            "ObjectName": "EntityServer.Core",
            "MethodName": "entityDetailsForLayer",
            "Args":[
                "L15T3C",
                DICT_SECTIONS[section],
                entname,
                "f413ebba23f7c8aeb682e22d5a9863f9"
            ],
            "SessionID":-1
        }

        i = 0
        while i < 10:
            res = requests.post(BASELINK, data=json.dumps(payload), headers=HEADERS).json()
            if 'entities' in res['Result']:
                break
            else:
                i += 1

        if i == 10:
            raise Exception("Problème d'extraction get_entity_table")

        for c in LIST_COLS:
            dict_entities[c] += [ent.get(c, pd.NA) for ent in res['Result']['entities']]
        dict_entities['Entities'] += [entname]*len(res['Result']['entities'])

    df_entities = pd.DataFrame(dict_entities)

    conv_name = df_entities.set_index('id')['name'].to_dict()

    df_entities.to_excel(f'section_{section}_entities.xlsx')

    dict_entities_req = {r.id: ['L15ST3C', 10] for i, r in df_entities.iterrows()}

    return df_entities, conv_name, dict_entities_req
```

Data

Fonction pour extraire les données de tassements

Entrée [43]:

```
def get_data(dict_entities_req):
    payload = {
        "Format": "GCFMessage",
        "Version": "0.1",
        "Timestamp": "2020-11-03T14:40:27.438Z",
        "Type": 262,
        "GUID": "b65f2a19-f2d2-4095-9a5a-3bfb374951b1",
        "MessageID": 4,
        "ObjectName": "WebSessionService.ProjectHelper",
        "MethodName": "getGraph",
        "Args": [
            {
                "graphType": "timeGraph",
                "graphEntities": dict_entities_req,
                "graphConfig": {
                    "dataSeriesType": "Detailed",
                    "startDate": "2000-01-01T00:00:00.000Z",
                    "endDate": "2030-31-12T23:59:59.999Z"
                }
            },
            "f413ebba23f7c8aeb682e22d5a9863f9"
        ],
        "SessionID":-1
    }

    i = 0
    while i < 10:
        res = requests.post(BASELINK, data=json.dumps(payload), headers=HEADERS).json()
        if res['Result']['errorCode'] == 0:
            break
        else:
            i += 1

    if i == 10:
        raise Exception("Problème d'extraction get_data")

    # write_to_pretty_json(res.json())

    dict_series = {}
    for i, elem in enumerate(res['Result']['data']['series']):
        ser = {
            dt.datetime.fromtimestamp(date//1000): val
            for date, val in elem
        }
        ser_name = conv_name[res['Result']['settings']['result']['entitySettings'][i]['entityId']]
        dict_series[ser_name] = pd.Series(ser)

    df_final = pd.DataFrame(dict_series)
    df_final.to_excel(f'section_{section}_resultats.xlsx')
```

Entrée [44]:

```
for section in ['TR1D', 'TR1F', 'TR1G', 'TR1H', 'TR2A', 'TR2B', 'TR2C', 'TR2D', 'TR2E', 'TR2F', 'TR2G', 'TR3A', 'TR3B']:
    print(section)
    list_entnames = get_layer_list(section)
    df_entities, conv_name, dict_entities_req = get_entity_table(section, list_entnames)
    get_data(dict_entities_req)
```

Application des fonctions et enregistrement sous format xlsx

section_TR1A_resultats.xlsx 12 863 Ko

section_TR1A_entities.xlsx 37 Ko

	118-D045_BAG_B01_08Z	118-D045_BAG_B01_18Z	220-D027_BAG_B02_010Z	220-D016_BAG_B02_09Z	220-D0
2017-06-14 15:06:48			3.38	1.849	
2017-06-14 22:02:30			2.35	1.069	

	id	x	y	name	
0	1.14E+08	1650006	8178299	118-D045_BAG_B01_08Z	201
1	1.14E+08	1650006	8178299	118-D045_BAG_B01_18Z	202

Code d'implémentation de la base de données

B

```sql
# PROJET
CREATE TABLE projet (
    id_projet VARCHAR(20) NOT NULL PRIMARY KEY,
    nom VARCHAR(255) NOT NULL,
    chef VARCHAR(255) NOT NULL,
    description VARCHAR(255)
)

# SECTEUR
CREATE TABLE secteur (
    id_secteur SERIAL PRIMARY KEY,
    id_projet VARCHAR(20) NOT NULL,
    code VARCHAR(50) NOT NULL,
    nom VARCHAR(500) NOT NULL,
    description VARCHAR(100000),
    longueur NUMERIC(10),
    pk_debut NUMERIC(25) NOT NULL,
    pk_fin NUMERIC(25) NOT NULL,

    CONSTRAINT fk_projet_secteur
    FOREIGN KEY (id_projet)
    REFERENCES projet(id_projet)
)

# ANNEAU
CREATE TABLE anneau (
    id_anneau SERIAL PRIMARY KEY,
    id_projet VARCHAR(20) NOT NULL,
    id_secteur INTEGER NOT NULL,
    pk INTEGER NOT NULL,
    troncon VARCHAR(20),
    pm INTEGER,
    anneau INTEGER,
    x NUMERIC(30,3),
    y NUMERIC(30,3),
    z_tn NUMERIC(30,3),
    z_base_tunnel NUMERIC(30,3),
    z_voute_tunnel NUMERIC(30,3),

    CONSTRAINT fk_secteur_anneau
    FOREIGN KEY (id_secteur)
    REFERENCES secteur(id_secteur)
)
```

```
44
45 # PARAMETRE_TUNNELIER
46 CREATE TABLE parametre_tunnelier (
47     id_anneau INTEGER NOT NULL PRIMARY KEY,
48     pk_rdc NUMERIC(30,3),
49     pm_rdc NUMERIC(30,3),
50     x_rdc NUMERIC(30,3),
51     y_rdc NUMERIC(30,3),
52     z_rdc NUMERIC(30,3),
53     date_debut_excavation TIMESTAMP,
54     date_fin_excavation TIMESTAMP,
55     date_debut_pose TIMESTAMP,
56     date_fin_pose TIMESTAMP,
57     "Avance_RDC[mm/mn]" NUMERIC(10,2),
58     "Vitesse_RDC[t/mn]" NUMERIC(10,2),
59     "Pas_Penetration[mm]" NUMERIC(10,2),
60     "Couple_RDC[kN.m]" NUMERIC(10,2),
61     "Puissance_RDC[kw]" NUMERIC(10,2),
62     "Energie[Mj/m3]" NUMERIC(10,2),
63     "PousseeTotale[kN]" NUMERIC(10,2),
64     "PousseeMolette[T]" NUMERIC(10,2),
65     "FrictionBouclier[kN]" NUMERIC(10,2),
66     "ForceContact[kN]" NUMERIC(10,2),
67     "TractionTrainSuiveur[kN]" NUMERIC(10,2),
68     "TractionJupe[kN]" NUMERIC(10,2),
69     "CapteurPressionsTerreFront1[bar]" NUMERIC(10,2),
70     "CapteurPressionsTerreFront2[bar]" NUMERIC(10,2),
71     "CapteurPressionsTerreFront3[bar]" NUMERIC(10,2),
72     "CapteurPressionsTerreFront4[bar]" NUMERIC(10,2),
73     "CapteurPressionsTerreFront5[bar]" NUMERIC(10,2),
74     "CapteurPressionsTerreFront6[bar]" NUMERIC(10,2),
75     "CapteurPressionsTerreFront7[bar]" NUMERIC(10,2),
76     "CapteurPressionsTerreFront8[bar]" NUMERIC(10,2),
77     "CapteurPressionsTerreFront9[bar]" NUMERIC(10,2),
78     "CapteurPressionsTerreFront10[bar]" NUMERIC(10,2),
79     "CapteurPressionsTerreFront11[bar]" NUMERIC(10,2),
80     "CapteurPressionsTerreFront12[bar]" NUMERIC(10,2),
81     "DeltaCHarticulation[mm]" NUMERIC(10,2),
82     "DeltaCVarticulation[mm]" NUMERIC(10,2),
83     "DeltaCHguidage[mm]" NUMERIC(10,2),
84     "DeltaCVguidage[mm]" NUMERIC(10,2),
85     "PressionInjectionMortierA1[bar]" NUMERIC(10,2),
86     "PressionInjectionMortierA2[bar]" NUMERIC(10,2),
87     "PressionInjectionMortierA5[bar]" NUMERIC(10,2),
88     "PressionInjectionMortierA6[bar]" NUMERIC(10,2),
89     "QuantiteMortierInjecteeA1[m3]" NUMERIC(10,2),
90     "QuantiteMortierInjecteeA2[m3]" NUMERIC(10,2),
91     "QuantiteMortierInjecteeA5[m3]" NUMERIC(10,2),
92     "QuantiteMortierInjecteeA6[m3]" NUMERIC(10,2),
93     "QuantiteMortierInjecteTotale[m3]" NUMERIC(10,2),
94
```

```
    UNIQUE (id_anneau),  %relation 1 to 1

    CONSTRAINT fk_anneau_parametretunnelier
    FOREIGN KEY (id_anneau)
    REFERENCES anneau(id_anneau)
)

# CAPTEUR
CREATE TABLE capteur (
    id_capteur SERIAL PRIMARY KEY,
    id_anneau INTEGER NOT NULL,
    id VARCHAR(20),
    nom VARCHAR(50),
    layer_name VARCHAR(50),
    layer_type VARCHAR(50),
    x_capteur NUMERIC(30,3),
    y_capteur NUMERIC(30,3),
    d_axe NUMERIC(10,3),

    CONSTRAINT fk_anneau_capteur
    FOREIGN KEY (id_anneau)
    REFERENCES anneau(id_anneau)
)

# MESURE_CAPTEUR
CREATE TABLE mesure_capteur (
    id_mesure_capteur SERIAL PRIMARY KEY,
    id_capteur INTEGER NOT NULL,
    date TIMESTAMP,
    mesure NUMERIC(10, 3) NOT NULL,

    CONSTRAINT fk_capteur_mesurecapteur
    FOREIGN KEY (id_capteur)
    REFERENCES capteur(id_capteur)
)

# FORMATION
CREATE TABLE formation (
    id_formation VARCHAR(10) PRIMARY KEY,
    nom VARCHAR(100) NOT NULL,
    ordre INTEGER NOT NULL,
    couleur_r INTEGER NOT NULL,
    couleur_v INTEGER NOT NULL,
    couleur_b INTEGER NOT NULL,
    eon VARCHAR(100),
    era VARCHAR(100),
    periode VARCHAR(100),
    stage VARCHAR(100),
    serie VARCHAR(100)
)

```

```
# DEFINITION_PARAMETRE_MECANIQUE_FORMATION
CREATE TABLE definition_parametre_mecanique_formation (
    id_definition_parametre_mecanique_formation VARCHAR(50) PRIMARY KEY,
    unite VARCHAR(20),
    categorie VARCHAR(100),
    description VARCHAR(1000)
)

# PARAMETRE_MECANIQUE_FORMATION
CREATE TABLE parametre_mecanique_formation (
    id_parametre_mecanique_formation SERIAL PRIMARY KEY,
    id_secteur INTEGER,
    id_formation VARCHAR(10),
    "gamma_h[kN/m3]" NUMERIC(50,3),
    "gamma_d[kN/m3]" NUMERIC(50,3),
    "E_M[MPa]" NUMERIC(50,3),
    "p_l*[MPa]" NUMERIC(50,3),
    "alpha[]" NUMERIC(50,3),
    "E'[MPa]" NUMERIC(50,3),
    "E_CT[MPa]" NUMERIC(50,3),
    "E'_ur[MPa]" NUMERIC(50,3),
    "E_CT_ur[MPa]" NUMERIC(50,3),
    "c_u[kPa]" NUMERIC(50,3),
    "phi_cu[°]" NUMERIC(50,3),
    "c'[kPa]" NUMERIC(50,3),
    "phi'[°]" NUMERIC(50,3),
    "psi[°]" NUMERIC(50,3),
    "OCR[]" VARCHAR(100),
    "K0[]" VARCHAR(100),
    "k_h[m/s]" VARCHAR(100),
    "k_v[m/s]" VARCHAR(100),

    UNIQUE(id_secteur, id_formation),

    CONSTRAINT fk_secteur_parametre_mecanique_formation
    FOREIGN KEY (id_secteur)
    REFERENCES secteur(id_secteur),

    CONSTRAINT fk_formation_parametre_mecanique_formation
    FOREIGN KEY (id_formation)
    REFERENCES formation(id_formation)
)

# STRATIGRAPHIE
CREATE TABLE stratigraphie (
    id_stratigraphie SERIAL PRIMARY KEY,
    id_anneau INTEGER NOT NULL,
    id_formation VARCHAR(10) NOT NULL,
    id_parametre_mecanique_formation INTEGER NOT NULL,
    z_base NUMERIC(30, 3),
    z_toit NUMERIC(30, 3),
```

```
197     epaisseur NUMERIC (30,3),
198     couverture_base NUMERIC (30,3),
199
200     CONSTRAINT fk_anneau_stratigraphie
201     FOREIGN KEY (id_anneau)
202     REFERENCES anneau(id_anneau),
203
204     CONSTRAINT fk_parametre_mecanique_formation_stratigraphie
205     FOREIGN KEY(id_parametre_mecanique_formation)
206     REFERENCES parametre_mecanique_formation(
        id_parametre_mecanique_formation)
207 )
208
209 # NAPPE
210 CREATE TABLE nappe (
211     id_nappe SERIAL PRIMARY KEY,
212     id_anneau INTEGER NOT NULL,
213     nom VARCHAR(10) NOT NULL,
214     z_nappe NUMERIC(30, 3),
215
216     CONSTRAINT fk_anneau_nappe
217     FOREIGN KEY (id_anneau)
218     REFERENCES anneau(id_anneau)
219 )
```

Script B.1 Code SQL pour la création des tables dans la base de données

Types de jointures de tables

C

Plus de détails à propos des jointures de tables avec Postgres sont fournis par PostgreSQLTutorial (2023). A noter que dans le cadre de cette thèse, les jointures sont toujours inclusives.

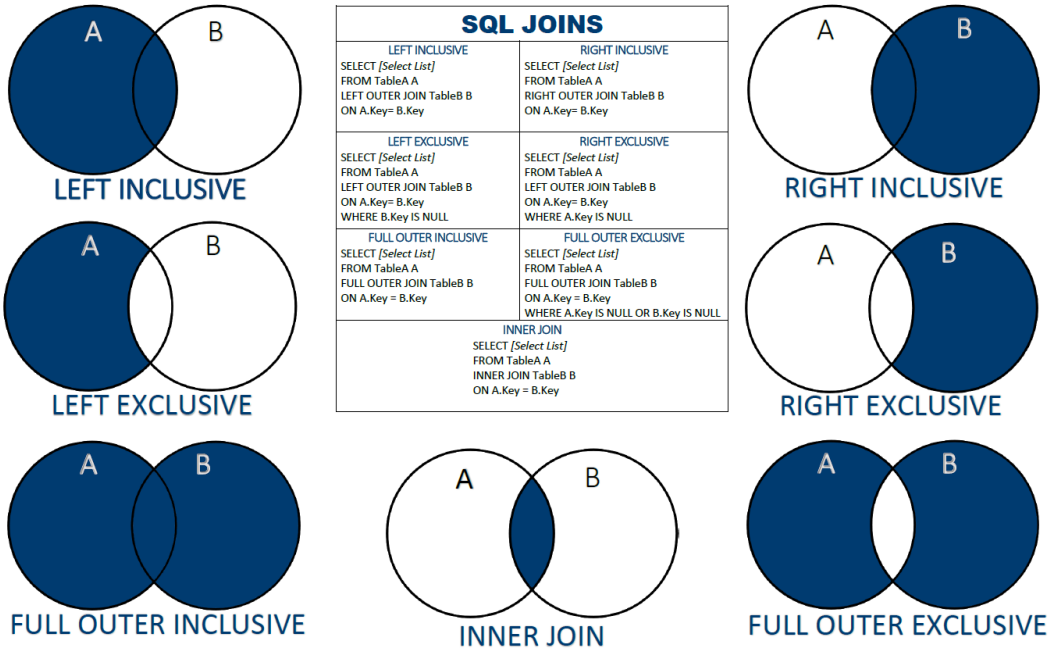


Figure C.1. Types de jointures SQL (U/ErockLobster, 2019)