UNIVERSITE MOHAMMED V DE RABAT
ECOLE MOHAMMADIA D'INGENIEURS
RABAT

Centre d'Etudes Doctorales
Sciences et Techniques pour l'Ingénieur

Thèse de Doctorat

Recours au Machine Learning dans le processus d'enquête mobile épidémiologique : cas de la Covid-19 avec RF et ODK

Présentée et soutenue le 06 octobre 2023 par

Loola BOKONDA

Pour obtenir le grade de

Docteur en Sciences et Techniques pour l'Ingénieur
Spécialité Informatique

Devant le jury composé de

Pr. Fatima-Zahra BELOUADHA	Présidente du jury	EMI-Rabat-UM5
Pr. Nissrine SOUISSI	Directrice de thèse	ENSMR
Pr. Khadija OUAZZANI TOUHAMI	Co-directrice de thèse	ENSMR
Pr. Abdessalam AIT MADI	Rapporteur	ENSA-Kénitra-UIT
Pr. Adil ANWAR	Rapporteur	EMI-Rabat-UM5
Pr. Saliha ASSOUL	Rapporteur	ENSMR
Pr. Maria LEBBAR	Examinatrice	ENSMR

À mon Père, Fidel Loola Esingi Ekambolongo, qui m'aime d'un amour sans égal (les actes sont plus éloquents que les mots) et m'accorde une confiance qui m'impose le sens de responsabilité. Ce père, autrefois enseignant de français, m'a communiqué le goût de la littérature, la philosophie ainsi que la rigueur de toujours vérifier la bonne orthographe et le bon sens des mots dans le dictionnaire en cas de doute. Bien que scientifique, ces qualités furent pour moi les prémices du profil de chercheur que je suis aujourd'hui. Puissiez-vous, Papa, trouver un peu de fierté dans cette modeste réalisation du premier de vos enfants.

À ma mère, Mathilde Mpembe Loola, qui a toujours cru en moi et qui, la première, m'a appelé Prof. Cette mère qui n'a ménagé aucun sacrifice pour me permettre d'aller aussi loin que possible. Dans l'espoir que ce diplôme de doctorat me permette d'obtenir de quoi vous accorder la protection que je n'ai pas pu vous procurer étant petit. Puissiez-vous être honorée par ce petit travail du premier fils de vos entrailles.

À mon épouse, Prisca Biadam Bokonda, celle que j'aime de tout mon être. Cette femme, merveilleuse et très intelligente, qui est devenue le centre de mon existence sur terre, l'étoile qui m'éclaire jour et nuit. Puisses-tu, Bae, récolter les fruits de tes immenses sacrifices. Cette thèse est autant la mienne que la tienne.

À mes futurs enfants, ces êtres que j'aime avant qu'ils existent ; j'essayerai de vous donner le même amour que j'ai reçu de votre grand père. Lorsque vous lirez ces lignes, je veux que vous puissiez garder en mémoire que ma thèse de doctorat ne constitue pas un défi qui vous oblige à faire mieux mais plutôt un message d'espoir, *un espoir : celui du possible*. Trouvez en cette thèse le courage de travailler sans relâche pour réaliser vos rêves les plus improbables. Labor omnia vincit improbus.

Aux générations futures du Congo RD et d'Afrique, ce travail est très petit et perfectible. Mais il se veut être porteur d'une part d'un message global, celui de nous faire prendre conscience que le développement de nos pays ne passera que par nous et par nous seuls, d'autre part d'une mission, celle de faire naître dans les cœurs des générations suivantes l'espoir du possible. C'est possible de développer l'Afrique. Ensemble bâtissons une Afrique meilleure. *L'espoir du possible.*

Remerciements

« *La première récompense de la réalisation d'un rêve est la satisfaction d'y être arrivé. Plus dure est le chemin, plus grand sera le souffle de soulagement* ». Bokonda L. Ce rêve m'aura coûté cinq ans de ma vie, sans compter les cinq ans de master. Durant ce parcours, j'ai rencontré des personnes sans qui ce travail n'aurait jamais été possible. J'aimerai les remercier.

Je réserve la primeur de ma gratitude à ma directrice de thèse, **Pr. Nissrine SOUISSI,** Professeur de l'Enseignement Supérieur (PES) à l'Ecole Nationale Supérieure des Mines de Rabat. J'ai rencontré Professeure Souissi pour la première fois, très exactement, le mardi 04 décembre 2018 dans la salle de l'équipe SiWeb lors de mon entretien avec tous les Professeurs de l'équipe. Dès les premières minutes, j'ai été frappé par la pertinence de ses interventions qui laissait entrevoir une intelligence aiguisée. J'ai, en ce moment précis, dans mon cœur désiré l'avoir comme directrice de thèse sans jamais formuler cela avec les mots. Cinq ans durant je n'ai pas été déçu, bien au contraire. J'ai découvert une Professeure très compétente, travailleuse mais aussi humble et patiente. C'est une grande fierté pour moi de vous avoir comme directrice de thèse. J'espère un jour être comme vous (Bien que je serai toujours l'élève et vous le maitre). Merci pour le temps et les efforts.

Mes remerciements les plus profonds et sincères à ma co-directrice de thèse, **Pr. Khadija OUAZZANI TOUHAMI**, Professeur Habilité (PH) à l'Ecole Nationale Supérieure des Mines de Rabat. Professeure Ouazzani, m'a appris à travailler avec une très grande rigueur scientifique. Ne laissant rien (ou presque rien) passer, Professeure Ouazzani est très minutieuse lorsqu'elle examine un article ou un chapitre. Cette rigueur m'obligeait constamment à chercher la qualité dans mes recherches et rédactions. Ses remarques sont toujours abondantes et très précises, même sur des aspects qui pourraient paraitre triviaux. Rien n'est laissé au hasard. J'ai aussi été encadré par une Professeure avec une grande empathie. Votre soutien moral dans le dernier sprint de la thèse m'était très précieux, vraiment très précieux. Sans vous, cette thèse n'aurait pas été possible. Merci pour tout.

J'espère de tout cœur que je garderai contact avec mes deux encadrantes, même après ma soutenance et que je pourrai vous recevoir chez moi au Congo, dans mon université.

Je tiens à remercier **Pr. Ounsa ROUDIES** Professeur de l'Enseignement Supérieur (PES) à l'Ecole Mohammadia d'Ingénieurs, cheffe de l'équipe SiWeb, l'équipe de recherche dans laquelle j'ai effectué ma thèse. Je garde en mémoire ces bons moments que nous avons passé à SiWeb, ces longues minutes de discussion sur la vie (ces anecdotes que vous me racontiez, etc.) ainsi

que vos précieux conseils. J'ai découvert en vous une personne avec des qualités humaines assez rares de nos jours. Merci Madame Roudies, merci.

Je remercie **Pr. Fatima-Zahra BELOUADHA**, Professeur de l'Enseignement Supérieur, de l'honneur que vous me faites d'être la présidente du jury de ma thèse. Merci de trouver en ces faibles mots toute mon estime et ma gratitude.

Mes vifs remerciements à la **Pr. Maria LEBBAR,** Professeur habilité, d'avoir accepté d'examiner ma thèse. Merci pour le temps et l'intérêt accordé à mes travaux de recherche.

Mes sincères remerciemments au **Pr. Abdsalam AIT MADI**, Professeur Habilité, d'avoir accepté d'être rapporteur de ma thèse. Votre prompt disponilité et implication m'ont permi d'enrichir mes travaux de thèse et d'avancer rapidement. Vos conseils et recommendations ont été ont amélioré la qualité de cette thèse. Merci.

J'exprime toute ma reconnaissance à la **Pr. Saliha ASSOUL**, Professeur Habilitée, d'avoir accepté d'être rapporteur de ma thèse. Votre vif intérêt pour mes travaux de recherche depuis quelques années m'a toujours encouragé. Merci aussi pour votre bienveillance à mon égard durant toutes ces années de ma thèse. Je garde en mémoire ces bons moments aux locaux de SiWeb ainsi qu'à l'ENIM.

J'exprime toute ma profonde gratitude au **Pr. Adil ANWAR**, Professeur de l'Enseignement Supérieur, d'avoir accepté d'être rapporteur de ma thèse. Votre implication et vos orientations m'ont permi d'améliorer la qualité de mes travaux. J'ai toujours été admiratif de votre energie et force de travail.

Merci à mon épouse, **Mme Prisca Bokonda**, experte en communication et gestion des projets internationaux qui a concrètement contribué à cette thèse par la traduction du français vers l'anglais et/ou la relecture de la quasi-totalité de mes articles de recherche et chapitres de ma thèse. Tes compétences linguistiques et managériales m'ont été d'une grande aide durant ce parcours. C'est bien de pouvoir compter en tout point sur son épouse. Merci mon aide semblable.

Merci à mon ami et petit frère de cœur, **Ir. Dior Masrné Reoukadji**, ingénieur systèmes. Quand tu es, à nouveau, entré dans ma vie en avril 2021 je ne savais pas que tu allais devenir l'un des éléments clefs de la réussite de ma thèse. Ton soutien inconditionnel m'a été d'une grande aide. Je ne saurai citer ce que tu as fait pour moi durant ces années, tellement la liste est longue ! Nous avons encore beaucoup de choses à réaliser ensemble. Merci pour tout, Masrané !

Merci à mon amie, **Pasteure Keren Leigh Zimmerman** pour sa contribution dans la traduction du français vers l'anglais et/ou la relecture de tous mes articles de recherche ainsi que le soutien moral durant ma thèse.

Merci à **M. Moussa SIDIBE** pour votre participation dans l'implémentation de la contribution principale de cette thèse. Vos compétences et expertises ont eté d'un grand apport.

Merci à mon meilleur ami, mon ami d'enfance, **Dr. Mbuyi Muenyi**, médecin spécialiste en Néphrologie. Merci pour ton soutien constant et insistant durant toute ma thèse et même avant. Tu étais toujours là pour me rappeler que c'était notre objectif depuis l'enfance. Merci aussi d'avoir relu mes chapitres de thèse pour me donner une vision médicale de mon travail. Notre amitié m'est précieuse. Merci pour ta loyauté.

Merci à mon meilleur ami, **Dr. Nguema Ongone**, professeur et docteur en chimie. Merci de l'exemple que tu as été pour moi en me précédent sur ce chemin d'une thèse de doctorat au Maroc. Merci pour ton soutien constant et d'avoir relu certains de mes chapitres. Je suis heureux d'avoir de tels amis. Merci pour ta loyauté.

Merci à mes petites sœurs **Dr. Xolali SILI** et **Ir. Evrard Adjallah** pour votre amour, votre soutien moral, vos prières, et nos moments de détente durant ces années.

Merci à **Mme Armela KENEHO AIDOO** pour votre aide et votre disponibilité constante.

Merci à l'entreprise **Kavaa Global Services**, et en particulier à son directeur **M. Olivier Jan Sonkeng,** de m'avoir reçu en son sein pour un stage de recherche dans le cadre de ma thèse mais aussi et surtout pour l'immense confiance que vous m'avez accordé durant toute la période de notre collaboration.

Merci à toutes ces **marocaines et marocains** qui ont été bienveillants envers moi durant ces onze années académiques. Je pense à mes **prof**., **camarades** de classe, **collègue**s de travail, sans oublier les **serveurs (es)** des cafés. Merci pour vos sourires, l'aide et la bienveillance !

Merci à mon petit frère biologique **Ir. Rodrigue Nkoloma Loola**, ingénieur métrologue et responsable technique. Merci pour ton soutien moral et spirituel. Être avec toi au Maroc m'a permis de me sentir moins seul.

Enfin, que toute personne qui a contribué, de près ou de loin, à la réussite de cette thèse trouve en ces faibles mots toute ma reconnaissance.

Résumé

Covid-19, Ebola, H1N1 sont des noms des épidémies/pandémies qui ont pris de court, certaines régions voire l'ensemble du globe et ont causé plusieurs morts. A l'ère du numérique, la lutte contre les épidémies n'est plus laissée aux seules mains des professionnels de la santé. Les ingénieurs et chercheurs IT y prennent aussi une part active.

La large propagation des appareils mobiles a donné aux chercheurs et ingénieurs IT une piste très intéressante à exploiter dans la lutte contre les épidémies. C'est ainsi que plusieurs suites logicielles mobiles de collecte de données ont vu le jour et ont été utilisées dans plusieurs domaines dont la lutte contre les épidémies.

Force est de constater que bien que ces suites apportent des solutions aux limites de la collecte de données via les formulaires en papier, elles sont elles-mêmes limitées dans l'analyse de données. En effet, ces suites ne proposent qu'une simple analyse statistique des données collectées.

Cette thèse de doctorat propose de renforcer dans l'une de ces suites l'analyse en utilisant les méthodes de l'intelligence artificielle en l'occurrence le Machine Learning. Accordant un intérêt particulier à la situation des pays en voie de développement, plus particulièrement les pays subsahariens, nous nous sommes intéressés aux systèmes mobiles de collecte de données les plus utilisés dans ces pays.

La suite logicielle Open Data Kit (ODK) s'est démarquée des autres non seulement par sa popularité dans cette région du monde mais aussi par le fait qu'elle a déjà été utilisée à plusieurs reprises dans la lutte contre les épidémies. La méthode Random Forests (RF) quant à elle s'est distinguée des autres par la qualité de ses métriques dans les études épidémiologiques prédictives. Considérant ces deux faits, cette thèse, après avoir modifié et enrichi la méthode RF, propose d'insérer la version enrichie de RF nommée RF-EP dans la suite logicielle ODK afin de mettre en place une suite logicielle adaptée au contexte des pays de l'Afrique subsaharienne et comportant un module d'analyse prédictive.

Mots-clés : E-santé, Intelligence Artificielle (IA), Machine Learning (ML), Analyse de données, Collecte de données, Systèmes mobiles, Open Data Kit, ODK, ODK-X, Epidémies, Coronavirus, Covid-19, Afrique, Random Forest (RF).

Abstract

Covid-19, Ebola, H1N1 - these are just some of the epidemics/pandemics that have taken some regions, if not the whole globe, by surprise and caused many deaths. In the digital age, the fight against epidemics is no longer left to health professionals alone. IT engineers and researchers are also playing an active role.

The widespread use of mobile devices has given IT researchers and engineers a very interesting avenue to explore in the fight against epidemics. As a result, a number of mobile data collection software suites have been developed and used in a variety of fields, including the fight against epidemics.

It has to be said that, although these suites provide solutions to the limitations of data collection via paper forms, they are themselves limited in terms of data analysis. In fact, these suites only offer a simple statistical analysis of the data collected.

This doctoral thesis proposes to strengthen the analysis in one of these suites using artificial intelligence methods, in this case Machine Learning. With a particular interest in the situation of developing countries, especially those in sub-Saharan Africa, we focused on the mobile data collection systems most widely used in these countries.

The Open Data Kit (ODK) software suite stood out not only for its popularity in this part of the world, but also for the fact that it has already been used on several occasions in the fight against epidemics. The Random Forests (RF) method, on the other hand, stands out from the others for the quality of its metrics in predictive epidemiological studies. Considering these two facts, this thesis, after having modified and enriched the RF method, proposes to insert the enriched version of RF named RF-EP into the ODK software suite in order to set up a software suite adapted to the context of sub-Saharan African countries and including a predictive analysis module.

Keywords: E-health, Artificial intelligence (AI), Machine learning (ML), Predictive analytics, Data gathering, Mobile systems, Open data kit, ODK, ODK-X, Epidemic prediction, Coronavirus, Covid-19, Africa, Random Forest (RF).

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Abréviation	Description
ACC	Accuracy
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
COVID-19	Maladie à Corona Virus 2019
CSV	Comma-Separated Values
DT	Decision Tree
ECOC	Error Correcting Output Codes
EoC	Ensemble of Classifiers
ES	Exponential Smoothing
GNB	Gaussian Naives Bayes
GP	Genetic Programming
HCT	Hématocrite
HMM	Hidden Markov Model
IA	Intelligence Artificielle
IIT	Institut Indien de Technologie
IT	Information Technology/Technologie de l'information
ILI	Influenza-like Illness
IMCI	Integrated Management of Childhood Illness
IST	Infections Sexuellement Transmissibles
JPME	Java Platform Micro Edition
KML	Keyhole Markup Language
k-NN	K-Nearest Neighbor
LASSO	Least Absolute Shrinkage and Selection Operator
LR	Linear Regression, *Logistic Regression*
MCC	Matthew Correlation Coefficient
MEPS	Mobile Epidemiological Prediction System
ML	Machine Learning
NB	Naive Bayes
ODK	Open Data Ki
ODK-X	Open Data Kit X
OMS	Organisation Mondiale de la Santé
ONG	Organisation Non Gouvernementale
PCA	Principal Component Analysis

PPA	*Peste Porcine Africaine*
QR	Question de Recherche
RAM	Random Access Memory
RDC	République Démocratique du Congo
RF	Random Forest
RF EP	Random Forests for Epidemiological Prediction
RI	Random Input
RMSE	Root Mean Square Error
RSM	Random Subspace Method
SIDA	Syndrome d'Immuno déficience Acquise
SMS	Short Message Service
SN	Sensitivity
SP	Specificity
SVM	Support Vector Machine
VIH	Virus de l'Immunodéficience Humaine

Chapitre 1 : Introduction Générale

1.1. Contexte et problématique

Commencée en Chine, la pandémie COVID-19 (Maladie à corona virus 2019) (OMS, 2020) s'est très vite répandue dans le monde et nous a tous obligé à lui accorder la plus grande attention. Cet épisode de l'histoire contemporaine, nous a tous rappelé que ce qui arrive aux autres peut aussi nous atteindre ; par ailleurs le devoir d'humanisme nous oblige de compatir à la souffrance des autres.

La COVID-19, bien que la plus récente et l'une des plus marquantes, est loin d'être la première pandémie que le monde a connue. De 1346 à 1353, la peste noire connue sous son pseudonyme anglais Black Death a causé des dizaines de millions de morts dans le monde (Prabhu & Gergen, 2022), (Benedictow, 2004). Certains chercheurs estiment que la peste noire a tué le tier de la population Européenne de l'époque (Gottfried & Robert , 2010). La grippe de 1918 a causé environ cinquante millions de morts dans le monde (Taubenberger & Morens, 2006). Cette pandémie courte dans sa durée a été l'une des plus meurtrières de l'histoire. Elle est historiquement la première pandémie grippale. Elle est parfois appelée « La pandémie de grippe "espagnole" de 1918-1919 », du fait de sa première découverte en Espagne.

Nous pouvons dans le même ordre citer *la Variole du Nouveau Monde* en 1520 (New World Smallpox) qui a été causée par l'arrivé des européens en Amérique. Ces derniers ont importé en Amérique une maladie dont les autochtones ne possédaient pas de système immunitaire (Patterson & Runge, 2002). Finissons cette petite liste des pandémies par la peste de Justinien qui a causé entre 30 et 50 millions de mort entre 541 et 549 (Sabbatani, et al., 2012).

A une échelle inférieure, mais tout aussi grave, nous avons des épidémies. A la différence des pandémies, elles ne touchent qu'une région et se limitent à celle-ci (Intermountain, 2020). Les trois dernières décennies de notre époque ont vu paraitre plusieurs épidémies dans plusieurs régions du monde. Nous nous intéressons dans le cadre de cette thèse aux pays en voie de développement avec un intérêt soutenu pour les pays de l'Afrique subsaharienne.

Le climat tropical des pays de l'Afrique subsaharienne favorise le développement de plusieurs épidémies, certaines devenues endémiques. Le paludisme continue de faire des morts dans la quasi-totalité des pays de cette région, l'OMS (Organisation Mondiale de la Santé) estime qu'il s'y est produit entre 280 et 480 millions de cas cliniques chaque année et entre 1,5 et 2,7

millions de décès ; un enfant de moins de 5 ans sur vingt meurt chaque année d'une maladie liée au paludisme, environ 5 à 40 pour cent des malades atteints de formes graves de paludisme décèdent (Baudon, 2000). Cette épidémie devenue endémique est le problème de santé numéro un, de cette région.

Dans cette partie du continent sévissent aussi d'autres épidémies comme Ebola qui a tué plus de trois mille personnes depuis sa déclaration officielle à l'Est de la République Démocratique du Congo (RDC) (LeMondeAfrique, 2020). Il sied de noter que le virus Ebola a touché plusieurs pays d'Afrique subsaharienne, dont le Sénégal, le Nigéria, le Gabon, etc. avec un taux de létalité d'environ 50 % (OMS, 2018). Toujours en RDC, le nombre des enfants victimes de la rougeole est estimé à plus de six-mille-six-cents depuis janvier 2019 pour plus de cinquante mille cas recensés (Monde, 2020). Au Cameroun les conséquences de la choléra ne sont plus à démontrer ; en 2010 dix-mille-sept-cent-cinquante-neuf cas ont été recensés dans huit régions du pays et six-cent-cinquante-sept décès (Djomassi, et al., 2013). Encore récemment, la (RFI, 2022) a annoncé une résurgence de l'épidémie de la dengue en Côte d'Ivoire ; il s'agirait d'une cinquième épidémie de la dengue dans cette partie de l'Afrique subsaharienne avec trois-cent-quatre-vingts cas de janvier 2022, à octobre 2022 dont cent-quarante-huit potentiellement très graves. En Ethiopie, l'OMS (Organisation Mondiale de la Santé) avait confirmé en janvier 2022 l'apparition de l'épidémie à virus Chikungunya, qui avait à cette date enregistrée trois-cent-onze cas suspects (MORVAN, 2022).

Face à ces défis majeurs de santé publique, à ces fléaux qui apportent tristesse et désolation, l'émergence des nouvelles technologies n'apporte-t-elle pas un message d'espoir ?

Force est de constater que l'Afrique comme le reste du monde n'est pas épargnée par la forte utilisation des smartphones. Dans tous les pays, les smartphones se sont imposés comme le moyen de communication par excellence. Leurs prix de plus en plus bas favorisent l'acquisition de ces derniers par les populations des pays d'Afrique subsaharienne. Du plus jeune au plus âgé, difficile de trouver une personne sans smartphone, en zones urbaines ou rurales. D'ici 2025, le nombre des connexions à l'aide du smartphone en Afrique subsaharienne atteindra environ 678 millions soient 65 % de la population (LAFRICAMOBILE, 2022).

Souvent utilisés pour communiquer (appeler, envoyer/recevoir des messages, réseaux sociaux), écouter la musique ou se distraire de diverses manières, les smartphones en particulier ou les technologies mobiles en général (tablette, montre connectée y compris) peuvent-elles contribuer dans la lutte contre les épidémies ?

La lutte contre les épidémies repose en grande partie sur la capacité à collecter, stocker, visualiser et analyser les données. Ce qui permet de choisir la campagne de vaccination la plus adaptée pour lutter contre l'épidémie selon les régions les plus touchées ou la tranche d'âge la plus atteinte, ou alors déterminer le plan de traitement à adopter à l'échelle nationale et dans le meilleur des cas prendre les mesures pour prévenir une recrudescence.

Jadis, les données étaient collectées via des formulaires en papier. Mais cette méthode a vite montré ses limites. La collecte et l'assemblage des données en grande quantité provenant des régions géographiquement très éloignées sont fastidieux, et l'analyse desdites données est quasi-impossible. Le besoin de collecter et d'analyser des quantités continuellement croissantes de données a poussé les chercheurs et ingénieurs IT à trouver d'autres moyens de collecte. C'est là qu'est né l'idée de collecter les données via les appareils mobiles. Plusieurs suites de collecte de données mobiles ont été proposées au fil des années entre autres : KoBoToolbox (KoBoToolbox, 2021), Open Data Kit (ODK, 2023), PendragonForms (PendragonForms, 2021) et Open Data Kit-X (ODK-X, 2023).

Enfin, un dernier contexte qu'il est important de souligner est l'importance accordée au Machine Learning ces dernières années. Le Machine Learning, branche de l'intelligence artificielle, a fait l'objet d'un grand intérêt ces dernières décennies et tout particulièrement ces dix dernières années. Les analyses statistiques bien qu'efficaces ne peuvent permettre l'analyse prédictive. Pour lutter plus efficacement contre la propagation des épidémies, plusieurs chercheurs se sont ainsi tournés vers l'analyse prédictive à l'aide du Machine Learning ; citons pour exemples ces quelques études : (Tapak, et al., 2019), (Koike & Morimoto, 2018), (Buvana & Muthumayil, 2021), (BOKONDA, et al., 2023).

La propagation des épidémies en Afrique subsaharienne d'une part et l'expansion sans cesse de l'utilisation des smartphones ainsi que l'intérêt porté au Machine Learning d'autre part, tracent le tableau complet du contexte et de la problématique de cette thèse de doctorat.

1.2. Objectifs de la thèse

L'objectif principal de cette thèse est d'une part de fournir à la communauté scientifique un outil permettant une meilleure analyse prédictive des épidémies afin de lutter contre la propagation de celles-ci, d'autre part de proposer aux pays en voie de développement le recours à une suite logicielle mobile capable d'effectuer l'analyse prédictive des épidémies ainsi qu'une ébauche de l'implémentation de ladite suite.

Pour y arriver nous nous étions fixé les sous-objectifs suivants :

1) Identification de la suite logicielle mobile la plus adaptée au contexte des pays en voie de développement. Il existe plusieurs suites logicielles mobiles pour la collecte de données. Il est donc important de pouvoir déterminer laquelle répond au mieux aux besoins et contexte souvent contraignant des pays en voie de développement plus particulièrement des pays subsahariens. Dans cette quête, il sera nécessaire de déterminer quelques critères de sélection, mais aussi dans le meilleur des cas joindre à la documentation littéraire, une analyse empirique par l'implémentation de chacune des suites.
2) Identification de la méthode d'analyse prédictive la plus appropriée à la prédiction épidémiologique. De nos jours, plusieurs méthodes d'analyse sont disponibles et utilisées. Très souvent, les utilisateurs se contentent d'opter pour une méthode parce que c'est la tendance sans forcément considérer le domaine où elle est souvent appliquée. Pour éviter de tomber dans le même piège, il nous a paru impérieux de procéder à une minutieuse étude de la littérature pour identifier la méthode d'analyse prédictive, en l'occurrence une méthode de Machine Learning, qui a fait ses preuves dans la prédiction des épidémies.
3) Amélioration de la méthode qui aura été identifiée. Le but n'est pas seulement d'identifier la méthode la plus adaptée à l'analyse épidémiologique mais aussi et surtout de l'améliorer afin de contribuer modestement à faire progresser la lutte contre les épidémies dans les pays d'Afrique subsaharienne.
4) Intégration de l'analyse prédictive dans la suite logicielle identifiée : modèle et début d'implémentation. Il s'agit ici de se fixer l'objectif de pourvoir proposer l'architecture d'une suite logicielle mobile dotée d'un module d'analyse prédictive spécialement conçu pour les épidémies. La partie suite logicielle devrait apporter le caractère d'adaptation au contexte des pays en voie de développement, alors que la partie analyse prédictive le coté d'adaptation à l'analyse des épidémies. Ce sous-objectif rassemblera les résultats obtenus par l'atteinte des sous objectifs précédents. Conscients des limites des moyens à notre disposition, nous n'avons pas l'ambition de fournir une implémentation complète clef en main de ladite suite logicielle mais plutôt toutes les briques nécessaires à la mise en place d'une telle suite.

Ainsi, tous les travaux de cette thèse tourneront autour de ces quatre sous-objectifs que nous venons de détailler.

1.3. Contributions de la thèse

Les contributions de cette thèse sont liées et complémentaires. Chacune d'elles, seule ou avec une autre, répond à un sous-objectif défini dans la section précédente ou directement à l'objectif principal défini dans la même section.

Cette thèse de doctorat comporte cinq contributions, les deux dernières étant les principales :

- **Etat de l'art et Etude benchmark des suites logicielles mobiles :** Cette contribution fournit à la communauté scientifique une étude relativement récente, empirique et littéraire sur plusieurs suites logicielles mobiles. Pour mener à bien cette étude, nous avons d'une part consultée la littérature à propos de chacune de ces suites, mais aussi nous les avons testés nous-même. Les suites logicielles ont été comparées sur la base des critères résumés par la Figure 1.1. Cet état de l'art a été publié dans (Bokonda, et al., 2020).

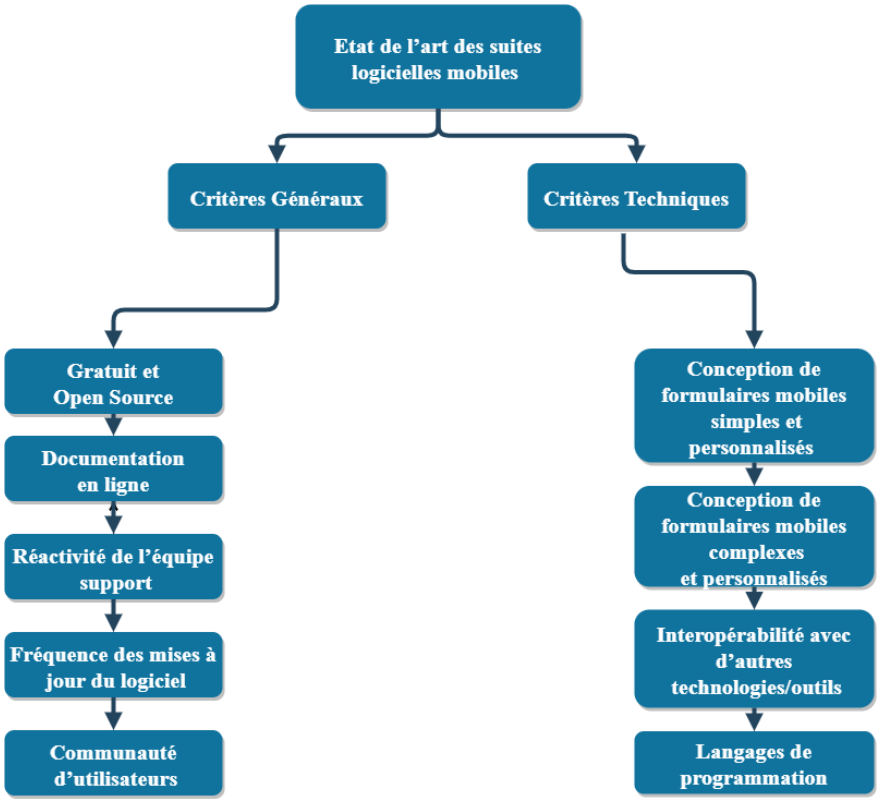


Figure 1.1: Critères de comparaison des suites logicielles mobiles

- **Etat de l'art et Etude benchmark de la suite logicielle ODK et de ses extensions :** Au travers de deux publications (BOKONDA, et al., 2019), (BOKONDA, et al., 2020) nous avons fourni à la communauté scientifique une documentation qui résume l'utilisation de ODK (Open Data Kit) dans plusieurs pays du monde et dans des

contextes et domaines différents. Ce qui permet de savoir à quel point et où ODK est populaire. Ces travaux nous ont aussi permis de déterminer les domaines dans lesquel ODK est le plus utilisé. Nous y avons aussi exploré les différents moyens d'implémentation de cette suite logicielle.

- **Etat de l'art des méthodes de Machine Learning :** Cet état de l'art a permis de définir les concepts et la terminologie utilisés en Machine Learning (ML), de déterminer les techniques de ML les plus utilisées par domaine d'apprentissage, les méthodes les plus fréquemment utilisées tous domaines confondus, les méthodes les plus utilisées en médecine, ensuite dans la prédiction des épidémies et pour finir nous avons pu choisir la méthode que nous avons améliorée par la suite. Les travaux de cette contribution ont été publiés dans (Bokonda, et al., 2020), (BOKONDA, et al., 2021).
- **Modèle ML enrichi :** La principale contribution de cette thèse consiste en ce modèle que nous avons conçu, implémenté, testé et comparé avec les autres. En effet, après avoir identifié la méthode ML la plus appropriée pour les prédictions épidémiologiques, nous l'avons enrichi afin qu'elle produise de meilleurs résultats que les autres dans la lutte contre la propagation des épidémies. Le nouveau modèle a été ensuite implémenté et testé sur un jeu de données de la COVID-19. L'algorithme que nous avons proposé et sur lequel se base notre modèle a été enrichi par les étapes suivantes : *la sélection des variables significatives, la normalisation des données, la réduction de dimension et la sélection de nouvelles variables qui synthétisent le mieux l'information dont l'algorithme a besoin sur la base du threshold_ratio préalablement défini*. Les travaux de cette contribution ont été publiés dans (BOKONDA, et al., 2023).
- **Système mobile de prédiction épidémiologique - MEPS (Mobile Epidemiological Prediction System) :** C'est le nom de la suite logicielle mobile étendue que nous proposons. Le MEPS comprend un module d'analyse prédictive spécialement conçu pour la prédiction des épidémies. Le MEPS est la fusion de la suite logicielle mobile ODK sélectionnée grâce aux précédents états de l'art et le modèle enrichi que nous avons proposé. Outre l'architecture, nous avons aussi proposé une ébauche d'implémentation.

1.4. Structure du mémoire de thèse

Ce mémoire rend compte du travail effectué tout au long de cette thèse. Il est structuré en huit chapitres.

Chapitre 1 définit le cadre général de la thèse. Il présente le contexte général dans lequel se sont déroulés les travaux de recherche, les objectifs poursuivis ainsi que les grandes lignes des contributions de cette thèse.

Chapitre 2 présente une comparaison de plusieurs suites logicielles mobiles de collecte de données. La comparaison est faite sur la base de la littérature ainsi qu'en tenant compte des résultats obtenus après implémentation de chacune des suites. L'objectif de cette comparaison est d'identifier la suite logicielle la plus appropriée pour la collecte de données dans les pays en voie de développement et plus particulièrement dans les pays subsahariens. Ce chapitre apporte la première partie du résultat escompté par le sous-objectif 1.

Chapitre 3 présente un état de l'art de la suite logicielle ODK. Après avoir été identifiée dans le chapitre 2 comme la suite logicielle la plus adaptée au contexte des pays en voie de développement, le chapitre 3 propose une étude minutieuse de ODK portant sur les différents lieux où elle a été utilisée (pays, villes, centre de santé, etc), les cas et domaines d'utilisation. Cet état de l'art propose aussi une architecture de la suite ODK qui permet d'explorer les différents moyens d'implémentation de celle-ci. Ce chapitre apporte la deuxième partie du résultat escompté par le sous-objectif 1.

Chapitre 4 présente un état de l'art des méthodes de Machine Learning pour l'analyse prédictive. Le chapitre oriente l'état de l'art vers la prédiction des épidémies afin de répondre au sous-objectif 2. Le chapitre présente aussi quelques statistiques des méthodes ML les plus utilisées par domaine d'application. Cet état de l'art a permis ainsi d'identifier la méthode ML la plus appropriées pour la prédiction des épidémies.

Chapitre 5 présente un formalisme détaillé de la méthode Random Forest (RF). Dans ce chapitre se trouvent les algorithmes de base sur lequel la méthode RF repose. Le but de ce chapitre est une meilleure compréhension de la structure et du fonctionnement profond de Random Forest pour pouvoir l'améliorer en la modifiant à partir de sa racine. Ce chapitre est un préambule à l'accomplissement du sous-objectif 3.

Chapitre 6 présente la version enrichie de la méthode Random Forest proposée dans cette thèse. Ce chapitre propose un formalisme théorique ainsi qu'une implémentation de ladite méthode enrichie. Le chapitre comporte aussi une comparaison des résultats obtenus avec la version améliorée et ceux obtenus avec les autres méthodes et la méthode RF par défaut. Ce chapitre apporte le résultat escompté par le sous-objectif 3.

Chapitre 7 présente un système mobile de prédiction épidémiologique (MEPS). Dans le chapitre se trouve l'architecture complète du MEPS, ainsi que son implémentation. Ce chapitre apporte un début de réponse au sous-objectif 4.

Chapitre 8 présente la conclusion de cette thèse. Le chapitre présente un résumé des résultats obtenus tout au long de cette thèse, analyse leurs limites et propose des voies d'amélioration comme travaux futurs.

Chapitre 2. Etude comparative des suites logicielles de collecte de données mobiles

2.1. Introduction

« Savoir pour prévoir afin de pouvoir. » Cette célèbre citation du philosophe français Auguste Comte énoncée au 19e siècle est plus vraie de nos jours qu'il y a deux siècles. Grâce à l'information, un pays peut procéder à une campagne de vaccination et éviter ainsi une grande épidémie ou alors améliorer l'efficacité des interventions des agents de santé. Grâce aux données collectées auprès des utilisateurs, une entreprise peut considérablement améliorer ses prestations ou lancer un nouveau produit. La collecte et la gestion des données sont devenues des activités inhérentes à l'émergence de tout type d'organisation (EL ARASS, et al., 2017) (EL ARASS & SOUISSI, 2018).

Si jadis les formulaires en papier furent le moyen privilégié pour collecter les données, l'avènement du numérique et la commercialisation du Smartphone en 2007 (PARK & CHEN, 2007) (BOKONDA, et al., 2019), ont poussé à délaisser cette méthode devenue archaïque. La collecte de données par appareils mobiles est devenue l'un des moyens les plus faciles et les plus utilisés par les organisations à travers le monde.

Mais le nombre sans cesse croissant d'outils de collecte de données par appareils mobiles ne facilite pas le choix aux Data Managers et à toute personne ou organisation s'intéressant à la collecte de données par appareils mobiles. Dans le but de fournir des informations claires, précises et vérifiées leur permettant de choisir les outils les mieux adaptés à leur contexte et besoin d'utilisation ; ce chapitre présente, évalue et compare quatre solutions logicielles de collecte de données par appareils mobiles :

- PendragonForms ;
- KoBoToolbox;
- Open Data Kit (ODK) ;
- Open Data Kit X (ODK-X).

Vu que nos travaux de recherche, (BOKONDA, et al., 2019) & (BOKONDA, et al., 2020), se focalisent sur la collecte de données par appareil mobile et la façon dont elle améliore le quotidien des populations, les solutions étudiées dans ce chapitre ont été choisies principalement grâce à leur large adoption, l'abondance de travaux de recherche disponibles, et pour trois d'entre elles, leur caractère humanitaire.

Ce chapitre est subdivisé de la manière suivante : la section 2 présente la méthodologie suivie pour analyser et comparer les quatre solutions logicielles. La section 3 décrit l'évolution, la

structure et le fonctionnement de chacune des solutions et fait une comparaison de ces quatre solutions en se basant d'abord sur des critères non techniques ; ensuite sur des critères techniques. La section 4 présente la synthèse d'une comparaison empirique de quatre suites sur base d'un formulaire conçu, implémenté et déployé ainsi qu'une visualisation de données. Les grandes caractéristiques et différences des suites logicielles étudiées y seront évoquées. Enfin, la section 5 donne comme conclusion de ce chapitre, les recommandations à suivre lors du choix de la solution à utiliser ainsi qu'une justification du choix de la suite effectuée dans le cadre de cette thèse.

2.2. Méthodologie

La comparaison effectuée dans ce chapitre suit une méthodologie de recherche bien structurée. Cette section cite et explique les étapes qui constituent cette méthodologie. Les quatre étapes constitutives de la méthode adoptée sont les suivantes :

- ***Collecte et Sélection des papiers :*** Pour chaque logiciel, nous avons collecté les documents qui permettent de mieux le comprendre et identifier ses fonctionnalités. La collecte de papiers s'est faite d'abord sur les sites web officiels (PendragonForms, 2021) , (KoBoToolbox, 2021), (ODK, 2023), (ODK-X, 2023) ensuite trois bases de données scientifiques ont été consultées : Google Scholar, SpringerLink et Science Direct. La recherche dans les moteurs de recherche et bases de données scientifiques a permis de rassembler les articles de recherche publiés par les concepteurs ou utilisateurs de chaque suite logicielle. La consultation des sites web a permis d'obtenir les documentations officielles fournies par les initiateurs de ces quatre solutions. La sélection s'est faite en tenant compte de l'apport du papier dans la compréhension de l'architecture ou de l'utilisation du logiciel mais aussi de son apport quant au traçage de l'historique évolutif de la plateforme.
- ***Lecture des documents et extraction de données :*** Les documents sélectionnés à l'étape précédente ont été minutieusement étudiés pour en extraire les éléments nécessaires à la bonne compréhension de ces logiciels. La section 3 présente les données extraites de ces documents.
- ***Installation des outils de la suite logicielle :*** Afin de tester chaque suite, les outils mobiles et desktop ont été installés. Ce processus d'installation des outils a permis de déterminer la complexité technique dans la mise en place des plateformes. Ce qui est un élément important dans le choix de la suite à utiliser car elle contribue à déterminer le

niveau technique exigé pour utiliser la suite. La section 3 donne les versions des outils installés pour chaque suite.

- ***L'exécution du workflow de collecte, stockage et visualisation de données :*** Pour chaque suite, un formulaire de collecte de données a été créé puis déployé sur une application mobile. Ensuite, nous avons procédé à une simple collecte de données, soumis les données collectées au serveur et finalement visualisé ces données avec l'outil proposé par la suite. Cette étape a permis de déterminer la complexité d'utilisation de chaque suite logicielle, élément déterminant dans le choix des logiciels à utiliser pour les organisations. La section IV présente les observations issues de l'application de ce processus. La figure 2.1 résume la méthode détaillée ci-dessus.

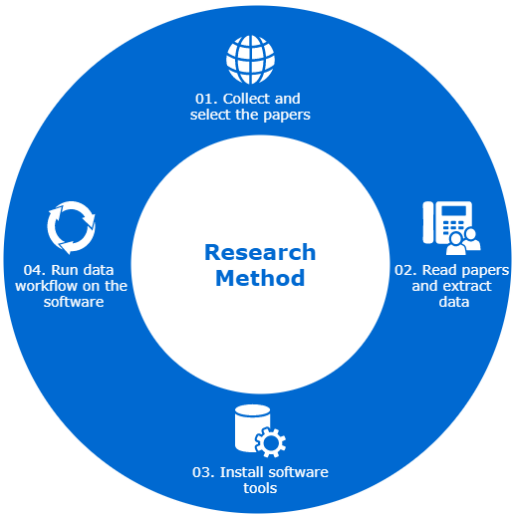


Figure 2.1 : Méthodologie de recherche

Cette méthode a conduit à une comparaison basée d'abord sur les ***fonctionnalités générales,*** détaillée dans la partie 2 de la section 3, qui est le résultat des deux premières étapes de la méthode. Ensuite, une comparaison basée sur les ***aspects* techniques** *(voir partie 3 de la section 3)*, et qui est le résultat des deux dernières étapes de la méthode.

2.3. Analyse comparative des suites logicielles de collecte de données.

La double contribution qu'apporte ce chapitre, à savoir d'une part le résultat des recherches sur les quatre solutions logicielles et d'autre part le résultat du test de chacune d'elles, nous a amené à adopter une démarche rigoureuse en quatre étapes. Cette démarche a permis de comparer les quatre solutions en tenant compte non seulement des aspects techniques (difficulté ou facilité d'installation des suites logicielles, la création des formulaires, la synchronisation des données, etc.), mais aussi d'autres aspects tels que la documentation sur la suite, les cas

d'utilisations déjà réalisées avec cette suite, la communauté d'utilisateurs, le caractère open source ou commercial de la suite, etc.

2.3.1. Présentation des outils

Dans cette partie nous présentons les différents outils ainsi que les versions des plateformes utilisées.

a. PendragonForms

PendragonForms est l'une des marques commerciales de Pendragon Software Corporation. PendragonForms est un concepteur de formulaires et un logiciel de collecte de données mobiles (PendragonForms, 2021). La solution offre principalement un outil desktop de conception de formulaire et une application mobile de collecte de données. Ceci dit, l'architecture de PendragonForms a fort évolué au fil des temps, s'adaptant ainsi aux technologies présentes sur le marché.

En 2007, PendragonForms conçut la version 5.1 de la plateforme, dans cette version la collecte de données devait se faire sur un ordinateur portable ou sur le Palm Treo (smartphone en cette année-là). Ensuite, en 2010, apparait la version 6, dans cette version la collecte de données se faisait via un site web qui pouvait être utilisé tant sur un appareil mobile que sur un ordinateur. C'est en 2014, dans sa version 7 que PendragonForms met en place une application mobile de collecte de données adaptée aux smartphones de la nouvelle génération.

Dans le cadre de l'étude faisant l'objet de ce chapitre, la version utilisée est le PendragonForms 8. Dans cette version, la solution logicielle offre quatre fonctionnalités/services à savoir :

- ***Mobile Form Design :*** Il s'agit ici de créer les formulaires de base, de collecte de données mobiles, ayant une logique et un workflow ordinaire. Aucune compétence en programmation n'est requise.
- ***Custom Development :*** C'est le service de PendragonForms qui facilite la conception des formulaires mobiles personnalisés et complexes.
- ***Mobile Database Synchronization :*** Les données collectées par les appareils mobiles sont stockées par défaut dans la base de données cloud de Pendragon. Grâce à l'outil *Data Manager* de PendragonForms, les données peuvent par la suite être téléchargées sous format Excel ou être modifiées.
- ***Integrating Mobile Application :*** C'est l'option qui permet d'intégrer PendragonForms aux applications mobiles existantes. Pour les entreprises ou organisations qui ont déjà

d'autres applications mobiles, Pendragon donne la possibilité de faire communiquer ses applications existantes avec PendragonForms.

b. KoBoToolbox

KoBoToolbox est un outil open-source qui sert à collecter et analyser des données par sondage sur le terrain (ex. évaluation des besoins), en utilisant des appareils mobiles tels que les téléphones portables ou les tablettes. Il est particulièrement adapté aux environnements difficiles, notamment, des champs de l'humanitaire et des situations d'urgence (KoBoToolbox, 2021), (DarithChhem, 2019) KoBoToolbox a été développé par les membres de la Harvard Humanitarian Initiative (KoBoToolbox, 2021) (Steinberg, 2019).

Malgré nos multiples recherches, nos discussions avec l'équipe support de KoBoToolbox, nos messages sur le forum, la faible documentation de KoBoToolbox ne nous a pas permis de retracer l'historique évolutif de cette solution logicielle.

A ce jour KoBoToolbox propose trois grandes fonctionnalités assurées de manière complémentaire par trois outils :

- ***FormBuilder :*** La plateforme propose un outil graphique de création de formulaires. Cet outil permet entre autres de réutiliser les questionnaires et blocs de questionnaires existants ; de créer des formulaires avec la possibilité de remplir et valider les champs dans n'importe quel ordre ; de partager les projets avec les autres tout en définissant différents niveaux d'autorisation et d'importer/exporter des formulaires xls.
- ***Collect Data :*** Cette fonctionnalité est assurée par *KoBoCollect,* l'application mobile de la plateforme, ou par *Enketo*un formulaire web pouvant être utilisé sur n'importe quel navigateur. *KoBoCollect* et *Enketo* permettent la collecte de données avec ou sans connexion internet (online and offline).
- ***Analyze and Manage Data :*** La solution propose un outil web qui permet, de créer des rapports de synthèse avec des graphiques et des tableaux ; de visualiser les données collectées sur une carte (map) ; de grouper les données collectées (par exemple par genre ou niveau d'éducation) au sein d'un rapport ou sur une carte ; d'exporter les données sous format Excel, CSV, KML, ZIP (pour les média) et SPSS. Cette fonctionnalité est assurée conjointement par l'outil web et le serveur KoBoToolbox.

Pour cette étude, nous avons installé et utilisé *KoBoCollect v1.23.3k*. Pour la conception des formulaires, nous avons utilisé la version en ligne du concepteur de formulaires de KoBoToolbox.

c. Open Data Kit

Open Data Kit (ODK) est une suite d'outils open source permettant aux organisations de collecter et de gérer leurs données (ODK, 2023). Une description plus large de l'architecture d'ODK ainsi que de son fonctionnement a été faite et publiée dans (BOKONDA, et al., 2020), ensuite l'historique de son évolution et utilisation dans (BOKONDA, et al., 2019). Notons tout de même qu'à ce jour ODK propose une architecture subdivisant les outils en trois ensembles :

- ***Desktop Clients :*** Ce sont les outils à installer sur un ordinateur portable ou fixe. Ils sont au nombre de trois. *ODK Build* qui est l'outil graphique d'ODK permettant de créer les formulaires. *ODK XLSForm* destiné à la création de formulaires plus complexes que ceux d'ODK Build et à la conversion de fichiers Excel en format Xform pris en charge par les outils d'ODK. *ODK Briefcase* permet d'importer et d'exporter les formulaires sur les serveurs ODK.
- ***Mobile Client :*** ODK propose un seul outil mobile : *ODK Collect*. Il s'agit d'une application mobile compatible uniquement avec les appareils Android. *ODK Collect* permet d'utiliser les formulaires préalablement créés avec *ODK Build* pour collecter les données (Hartung, 2010).
- ***Servers :*** ODK propose deux serveurs. *ODK Aggregate,* il permet le stockage, l'analyse et la visualisation des données collectées par ODK Collect. C'est le serveur d'ODK le plus utilisé à ce jour. *ODK Central*, une alternative à ODK Aggregate prenant en charge les nouvelles technologies notamment l'API REST.

Pour cette étude, nous avons installé et utilisé *ODK Collect v1.24.1 et ODK Aggregate v1.7.3.* Pour la conception des formulaires, nous avons utilisé la version en ligne du concepteur des formulaires d'ODK.

d. Open Data Kit X

ODK-X anciennement appelé ODK 2.0, est une autre suite d'outils développé par les fondateurs d'ODK (Brunette, 2013) (Brunette, 2017). Comme ODK, ODK-X permet de faire la collecte, le stockage et l'analyse de données. A la différence de son prédécesseur, ODK-X permet de créer des formulaires pour des études ayant des workflows très complexes et exige des compétences techniques assez poussées pour son utilisation.

Une description plus large de l'architecture d'ODK-X ainsi que de son fonctionnement a été faite et publiée dans (BOKONDA, et al., 2020), ensuite l'historique de sa création, évolution et utilisation dans (BOKONDA, et al., 2019)

De même qu'ODK, ODK-X propose une architecture subdivisant les outils en trois ensembles :

- ***Desktop Clients :*** ODK-X propose deux outils desktop. *ODK Application Designer,* c'est l'équivalent d'ODK Build. Il permet de créer les formulaires qui seront utilisés par les applications mobiles d'ODK-X. *ODK Suitcase* c'est l'équivalent d'ODK Briefcase, il permet d'importer et d'exporter les formulaires sur les serveurs ODK-X (BOKONDA, et al., 2020)**.**
- ***Mobile Clients :*** Contrairement à ODK, ODK-X ne propose pas un outil mobile mais trois. *ODK Survey et ODK Tables* (Hong, 2011) sont des applications mobiles permettant de faire la collecte de données. *ODK Services* est installé comme un prérequis à l'utilisation des deux autres. Il permet la synchronisation de données entre toutes les applications ODK-X et gère le service d'accès à la base des données et aux fichiers (BOKONDA, et al., 2020)**.**
- ***Servers :*** ODK-X possède un serveur : *ODK SyncEndpoint*. Mais en outre, il est possible, pour tous les v1.x.x d'ODK Aggregate d'activer la configuration permettant la prise en charge des outils d'ODK-X. On parle alors d'ODK Aggregate Tables Extension comme un deuxième serveur d'ODK-X. Il sied de signaler que cette configuration a été supprimée dans les v2.x.x d'ODK Aggregate.

Pour cette étude, nous avons utilisé et lorsque nécessaire installé *ODK-X Services v2.1.4 rev 234 ; ODK-X Tables v 2.1.4 rev 234 ; ODK-X Application Designer v2.1.4 ; ODK Aggregate v1.7.3 (avec une configuration spéciale pour prendre en charge les outils d'ODK-X).*

2.3.2. Analyse basée sur les fonctionnalités générales

Cette partie traite des aspects non techniques mais essentiels lors du choix de la solution à utiliser pour la collecte des données mobiles. Le Tableau 2.1 donne, pour les quatre solutions, le résultat de cette évaluation. Les critères retenus sont :

- ***Gratuit et Open Source :*** Ceci doit être l'une des premières, si pas la première, information à avoir avant d'effectuer son choix. Ceci signifie que le logiciel peut être utilisé, modifié (code source accessible) et partagé sans payer, car sa conception est publique.

- ***Documentation en ligne :*** Savoir si le logiciel dispose d'une grande documentation sur internet permet de savoir d'avance si sa compréhension sera facile ou pas. Les logiciels les mieux documentés sont souvent les mieux compris.
- ***La réactivité de l'équipe support :*** Bien qu'un logiciel ait une bonne documentation en ligne, cela n'empêche que les techniciens ou ingénieurs aient besoin de contacter l'équipe support pour une aide technique. Pour cela il est préférable de savoir d'avance si cette équipe réagit rapidement ou pas.
- ***La fréquence des mises à jour du logiciel :*** Ceci permet de savoir si les bugs actuels du logiciel seront corrigés et/ou de nouvelles fonctionnalités ajoutées.
- ***Communauté d'utilisateurs :*** Ceci revient à savoir si le logiciel a déjà été utilisé par plusieurs personnes dans le monde.

Tableau 2.1: Les critères généraux et non techniques

(◆ : Très bien / Oui ; ◆ : Bien ; ◇ : Pauvre / Non)

	Pendragon Forms	KoBoToolbox	ODK	ODK-X
Gratuit et Open Source	◇	◆	◆	◆
Documentation en ligne	◆	◇	◆	◆
La réactivité de l'équipe support	◆	◇	◆	◆
La fréquence des mises à jour du logiciel	◆	◆	◆	◆
Communauté d'utilisateurs	◇	◆	◆	◆

La première information que le tableau 1 donne est essentielle. Cette information (*Gratuit et Open Source*) permet de comprendre que si l'utilisateur souhaite étendre la solution logicielle en y ajoutant une fonctionnalité spécifique et/ou s'il n'a pas un budget conséquent pour le projet, *PendragonForms* n'est peut-être pas la bonne solution. Il faudra s'orienter vers l'une des trois autres solutions.

Des quatre solutions, ODK est celle qui a la meilleure documentation. Ceci est dû au fait qu'ODK a une grande communauté d'utilisateurs, notamment dans les pays en voie de développement, mais aussi parce qu'il est open source et plus ancien que les deux autres (ODK-X et KoBoToolbox). Plusieurs ingénieurs et chercheurs ont rédigé des travaux expliquant une amélioration, une extension ou simplement l'utilisation d'ODK pour un cas précis.

La documentation d'ODK comporte :

- Un site web bien détaillé.
- Une documentation officielle en version pdf d'environ 600 (six cent) pages.
- De nombreux articles de recherche dans les bases de données scientifiques.
- De nombreuses vidéos sur YouTube qui montrent la procédure pour installer ou utiliser les outils d'ODK.

ODK-X en revanche, bien que comportant une documentation officielle aussi bien détaillée que celle d'ODK, n'a pas autant d'articles de recherche et vidéos sur YouTube que son prédécesseur. Ceci est justifié par le fait que ODK est plus ancien que ODK-X (conçu en février 2013, (Brunette, 2013)). KoBoToolbox a un site web ayant très peu d'informations et peu d'articles de recherche et vidéos sur YouTube et n'a pas de documentation officielle. PendragonForms possède un site web contenant suffisamment d'informations pour comprendre la suite ainsi qu'une documentation officielle en version pdf mais presque pas d'articles de recherche et vidéos sur YouTube.

Afin de tester la réactivité des équipes support, nous nous sommes inscrits aux forums de KoBoToolbox et ODK/ODK-X. PendragonForms n'a pas encore de Forum à ce jour, nous avons donc communiqué par emails avec l'équipe support. L'équipe support de PendragonForms nous a confirmé par email que le projet n'a pas de forum pour l'instant.

Nous avons constaté qu'un message publié dans le forum d'ODK/ODK-X recevait une réponse dans moins de 24 heures, de même pour un email envoyé à l'équipe support de PendragonForms. Tandis qu'un message publié dans le forum de KoBoToolbox ou envoyé par email à l'équipe support pouvait attendre jusqu'à plus d'une semaine avant de recevoir une réponse.

2.3.3. Analyse basée sur les aspects techniques

Cet axe d'analyse traite des aspects liés à l'installation et l'utilisation des outils des solutions sélectionnées ainsi que ceux liés à la création et la personnalisation des formulaires ou applications de collecte de données. Le résultat de cette évaluation est résumé dans le Tableau 2.2.

Les critères que nous avons retenus pour cette évaluation sont :

- ***Conception de formulaires mobiles simples et personnalisés*** : Ce critère évalue la facilité qu'il y a de créer des formulaires simples mais personnalisés.
- ***Conception de formulaires mobiles complexes et personnalisés*** **:** Ce critère évalue la facilité qu'il y a de créer des formulaires complexes et personnalisés.
- ***Interopérabilité avec d'autres technologies/outils*** : Ce critère évalue la possibilité d'utiliser les outils d'une solution avec les outils n'appartenant pas à celle-ci.
- ***Langages de programmation*** : Ce critère indique les langages de programmation utilisés pour développer les outils de la plateforme. Ces langages peuvent être utilisés pour étendre ces outils.

Tableau 2.2: Critères techniques

(◆ : critère rempli ; ◇ : Critère non rempli / non recommandé)

	PendragonForms	KoBoToolbox	ODK	ODK-X
Conception de formulaires mobiles simples et personnalisés	◆	◆	◆	◇
Conception de formulaires mobiles complexes et personnalisés	◆	◇	◇	◆
Interopérabilité avec d'autres technologies/outils	◆	◆	◆	◆
Langages de programmation	–	Java, JavaScript, Python	Java, JavaScript Python	Java, JavaScript

Les concepteurs graphiques des formulaires de KoBoToolbox, PendragonFormset ODK, rendent facile la création des formulaires. En revanche ces concepteurs sont très limités lorsqu'il s'agit de la création de formulaires complexes. PendragonForms couvre ce manque par la mise en place de services de développement personnalisé destinés à accompagner ceux qui veulent créer des formulaires complexes avec PendragonForms.

En revanche, ODK-X n'offre pas de concepteur graphique de formulaires. Le concepteur des formulaires d'ODK-X, permet de créer des applications mobiles de collecte (formulaires) en utilisant notamment les langages de programmation JavaScript et HTML. Ce qui rend ODK-X plus adapté à la création de formulaires complexes qu'à ceux qui sont simples.

De par leur caractère open source, les outils des trois solutions sont ouverts à une utilisation avec d'autres outils externes. PendragonForms, malgré qu'il soit une solution commerciale, permet aussi d'utiliser ces outils avec d'autres qui ne lui appartiennent pas. Il faudra tout de même noter que les outils d'ODK ne sont pas compatibles avec ceux d'ODK-X, sauf le serveur ODK Aggregate d'ODK qui peut être configuré pour prendre en charge les outils d'ODK-X. Cette configuration d'ODK Aggregate n'est possible que sur les v1. x.x.

Jusqu'à la v5.1, PendragonForm offrait un kit de développement pour les développeurs, qui permettait d'ajouter de nouvelles fonctionnalités à la plateforme en utilisant les langages de programmation C et C#. Mais dans les documentations des versions succédant la v5.1 ainsi que sur le site web officiel de la plateforme, il n'est aucunement fait mention de cela. Nous avons contacté par email l'équipe support à ce sujet et ils nous ont confirmé que PendragonForms n'offre plus ce service.

2.4. Synthèse

Cette section traite des résultats obtenus après le test des suites logicielles sélectionnées en allant de la création du formulaire jusqu'à la visualisation des données collectées. Le formulaire créé a pour objectif de faire un sondage d'opinion sur la vaccination afin de savoir comment le vaccin est perçu au sein de la population. Une telle étude pourrait permettre de savoir s'il est nécessaire de faire une campagne de sensibilisation et d'information avant de procéder à une campagne de vaccination dans un territoire donné.

Pour tester les quatre solutions logicielles, nous avons créé un formulaire avec cinq champs notamment :

- ***Full Name*** **:** Un champ de type 'text' donnant la possibilité à l'utilisateur de remplir son nom complet.
- ***Age*** **:** Un champ de type 'number' ne prenant que des numériques, pour le remplissage de l'âge.
- ***Gender*** **:** Un champ de type 'radio' donnant la possibilité de faire un seul choix entre les deux options proposées ('Female' et 'Male').
- ***Have you ever been vaccinated ?*** **:** Un champ de type 'radio' donnant la possibilité de faire un seul choix entre les deux options proposées ('Yes' et 'No').
- ***Do you trust vaccination ? Why ?*** **:** Un champ de type 'text' permettant d'avoir l'avis de la personne interrogée sur la vaccination.

2.4.1. KoBoToolbox et ODK

Ces deux solutions ont en commun le fait qu'elles offrent un outil graphique pour la conception des formulaires mais ne permettent pas de créer des formulaires complexes.

KoBoToolbox offre un outil en ligne (site web) qui comporte à la fois le concepteur des formulaires et le serveur. KoBoToolbox a un fonctionnement à base de projets. Pour créer un questionnaire, il faut d'abord configurer un projet. Le projet ainsi créé peut être partagé publiquement ou alors, à l'aide des adresses emails, spécifier les utilisateurs qui auront le droit d'accès à ce projet. La Figure 2.2 montre le concepteur des formulaires de KoBoToolbox avec le projet qui a été créé pour cette étude. Sur l'application mobile (KoBoCollect), pour avoir accès aux formulaires, il faut spécifier l'url du serveur ainsi que le nom d'utilisateur et mot de passe.

La Figure 2.3 présente un formulaire entièrement rempli avec KoBoCollect. Après collecte et soumission des données au serveur, la visualisation des données se fait sur le site web qui se trouve sur le même serveur que le designer des formulaires.

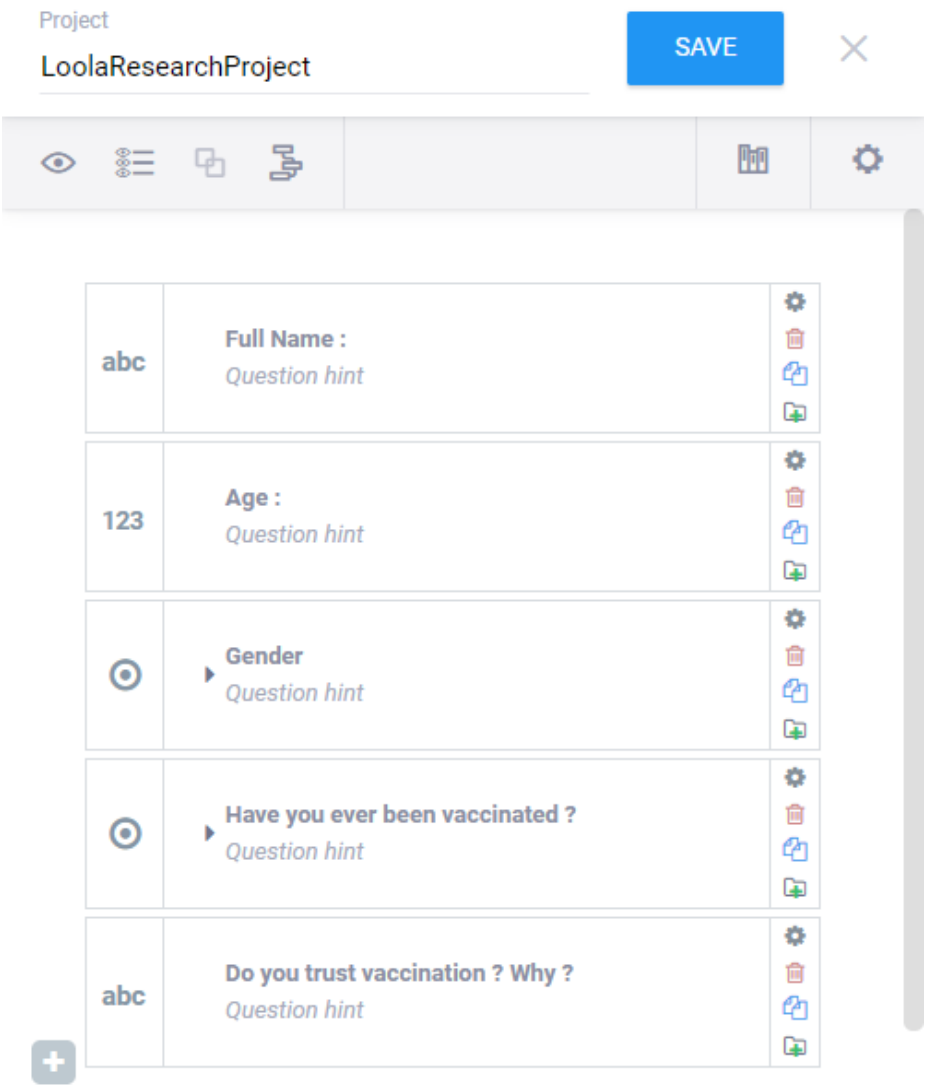


Figure 2.2 : KoBoToolbox Form Designer

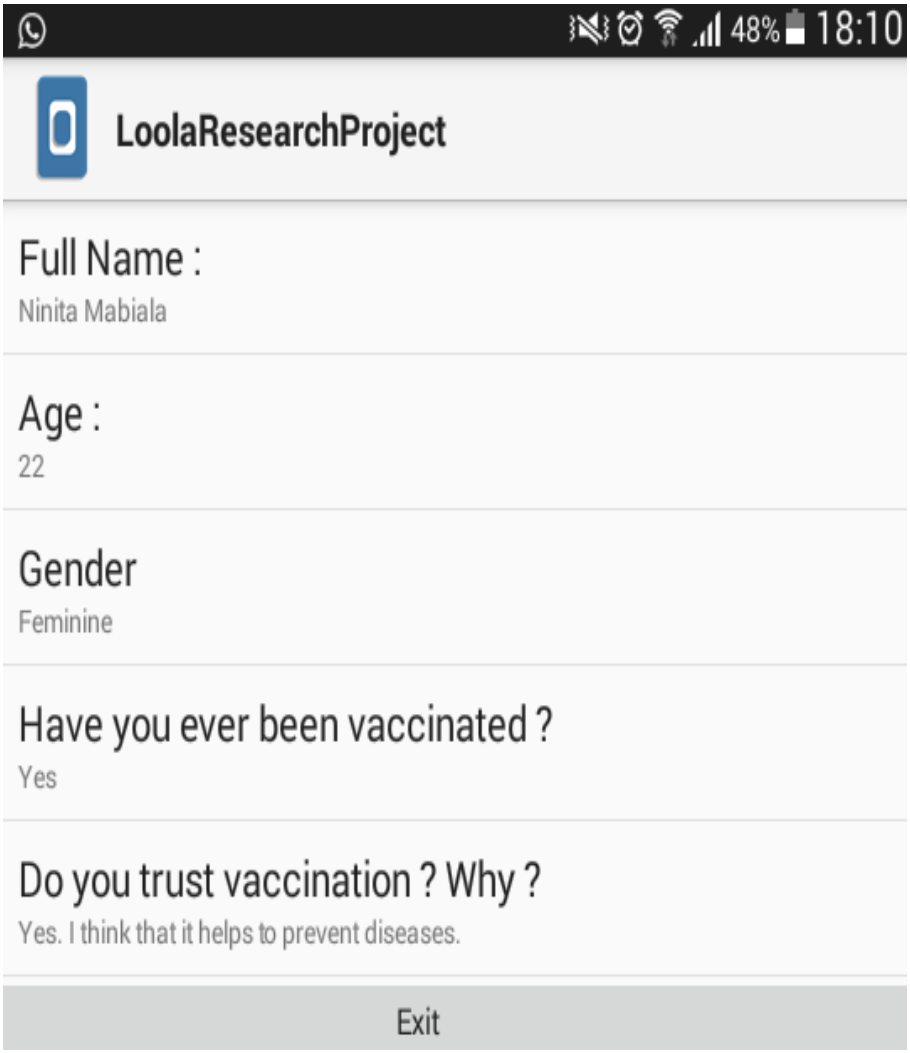


Figure 2.3 : KoBoToolbox - KoBoCollectMobile App

La Figure 2.4 illustre la visualisation des données sur le serveur de KoBoToolbox.

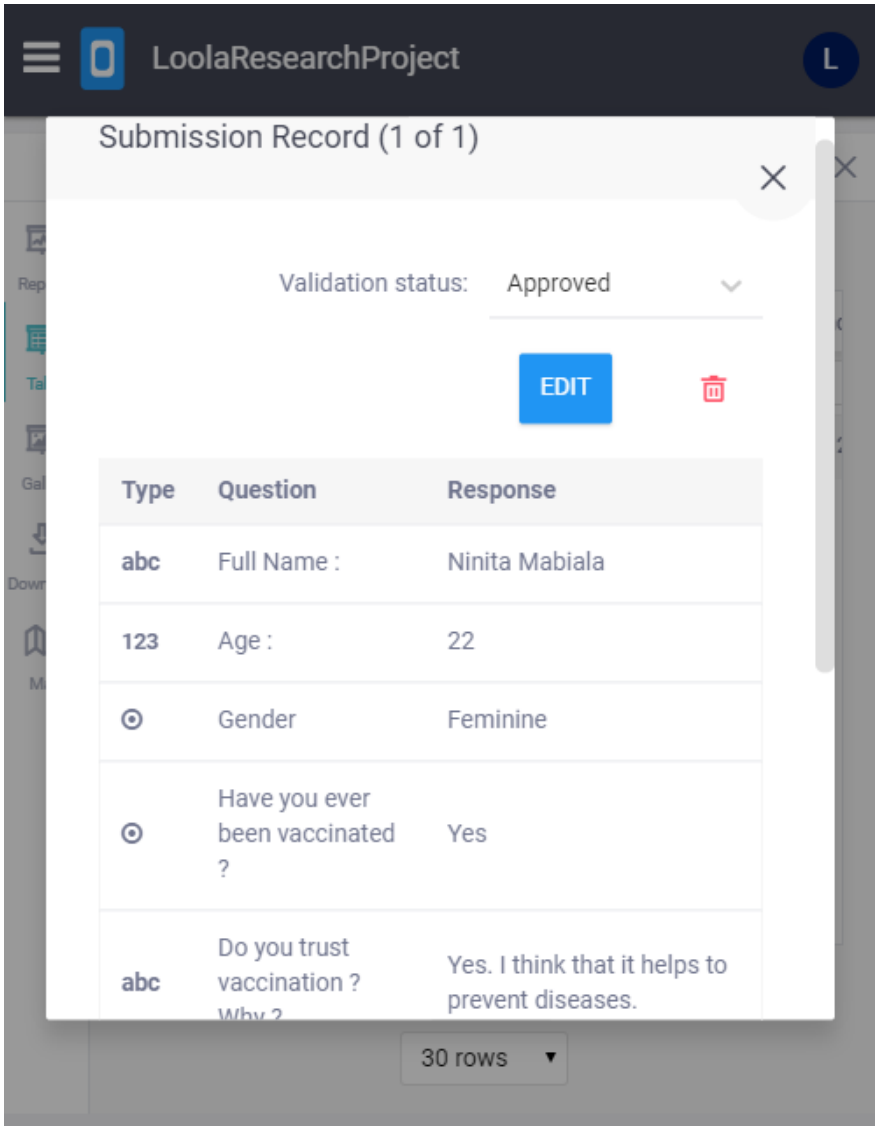


Figure 2.4 : KoBoToolboxvisualisation of collected data

Le fonctionnement de KoBoToolbox ainsi décrit est le même pour ODK. A la différence qu'ODK sépare le concepteur des formulaires (ODK Build), du serveur (ODK Aggregate).

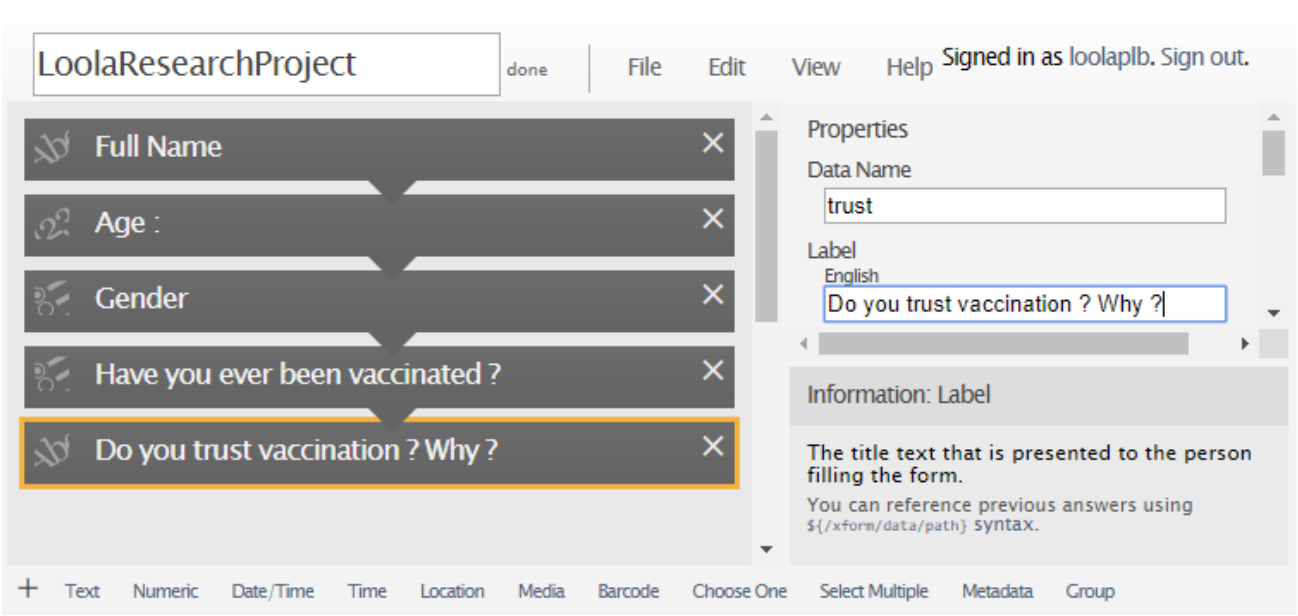


Figure 2.5 : ODK - ODK BuildInterface

La Figure 2.5 présente ODK build avec le formulaire que nous avons créé pour cette étude pendant que la Figure 2.6 présente une seule étape dans la collecte de données avec ODK Collect.

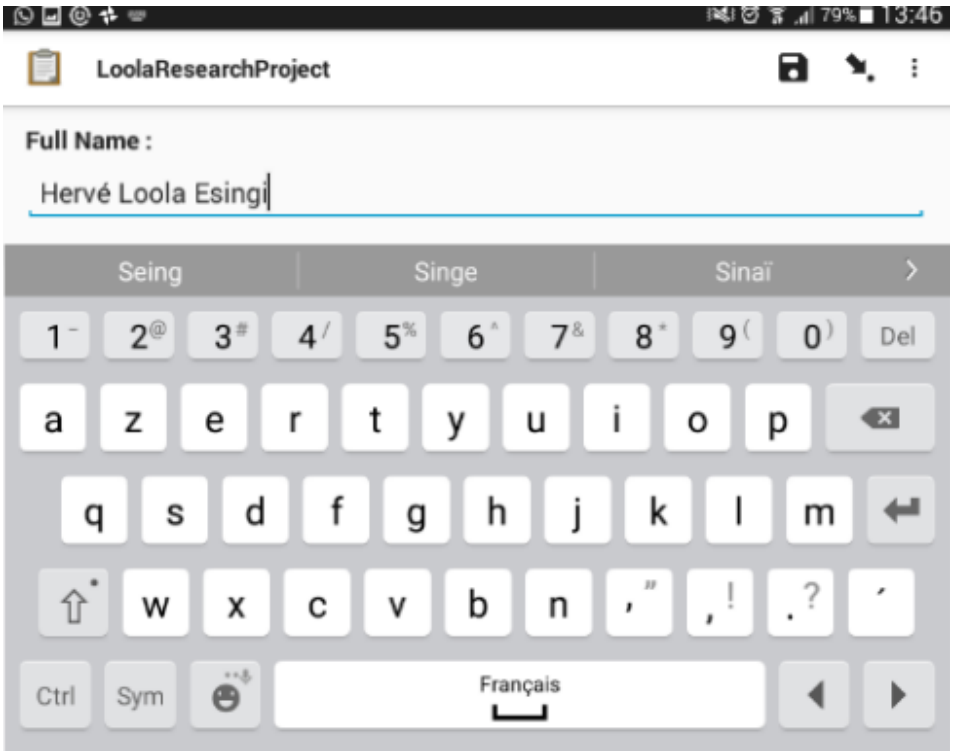


Figure 2.6 : ODK - ODK Collect Mobile App

La Figure 2.7 présente ODK Aggregate ayant les données collectées. Pour utiliser correctement ODK, il faudra configurer ODK Aggregate sur une plateforme Cloud (Google, AWS, etc.) ou héberger le projet sur son propre serveur ou machine virtuelle. Ce qui fait qu'ODK nécessite plus de compétences techniques que KoBoToolbox pour commencer à l'utiliser. La documentation ODK fournit des didacticiels détaillés étape par étape pour toutes ces options afin de minimiser les connaissances techniques requises pour la configuration du serveur.

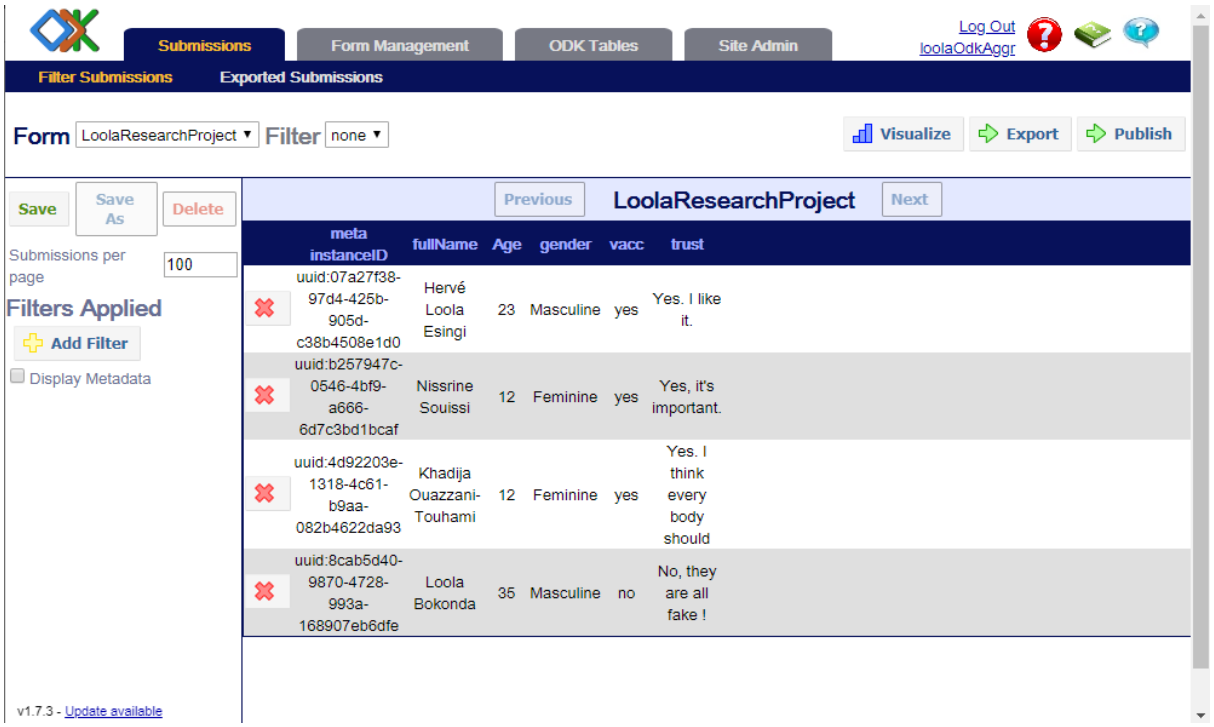


Figure 2.7 : ODK - ODK Aggregate interface

S'agissant des applications mobiles, ODK Collect et KoBoCollect sont très proches, parce que KoBoCollect a été créé en se basant sur ODK Collect. Les deux applications mobiles sont tellement proches que lorsqu'on veut collecter les données à partir d'un formulaire créé avec ODK Build, Android propose de choisir l'application à utiliser entre ODK Collect et KoBoCollect. Un formulaire créé avec le concepteur de formulaire d'ODK (ODK Build) peut être utilisé sur la plateforme KoBoToolbox et réciproquement. KoBoCollect et ODK Collect peuvent être téléchargés à partir de la Play Store Google.

2.4.2. PendragonForms et ODK-X

Ces deux plateformes ont en commun le fait qu'elles offrent la possibilité de créer des formulaires ayant des workflows non-linéaires et plus complexes. PendragonForms offre tout cela par l'assistance d'une équipe destinée à accompagner les clients et ODK-X par la performance de ses outils dont son concepteur de formulaires (ODK-X Application Designer)

illustré à la Figure 2.8. Par exemple, ODK-X donne la possibilité de lire les données d'une source structurée (par exemple, les fichiers CSV) et utiliser les résultats comme des options de réponse ou pour mettre en place une certaine logique dans l'exécution d'un formulaire, un autre exemple c'est le lancement des sous formulaires qui stockent les données dans différentes tables (Steinberg, 2019) (Steinberg, et al., 2019). A ce niveau, la différence entre les deux réside dans le fait qu'ODK-X exige suffisamment de compétences techniques, notamment en programmation des logiciels, alors que PendragonForms en exige que très peu du côté du client (entreprise ou organisation qui utilise la solution logicielle).

Un autre point commun entre les deux solutions est, d'une part, qu'elles ont une bonne documentation en ligne, d'autre part une équipe support très réactive pour PendragonForms et un forum très actif pour ODK-X. Ceci aide énormément à atténuer l'exigence de compétences techniques avec ODK-X. En outre, les deux solutions donnent la possibilité d'intégrer leurs outils dans un écosystème logiciel existant. Donc la question de la compatibilité avec les équipements présents dans l'entreprise ne se pose pas.

Quant aux outils proposés et au fonctionnement, PendragonForms est très proche de KoBoToolbox parce qu'il offre un site web contenant à la fois les concepteurs graphiques des formulaires illustrés à la Figure 2.9 et le serveur illustré à la Figure 2.10 ainsi qu'une seule application mobile de collecte de données que l'on peut télécharger à partir de la Play Store Google.

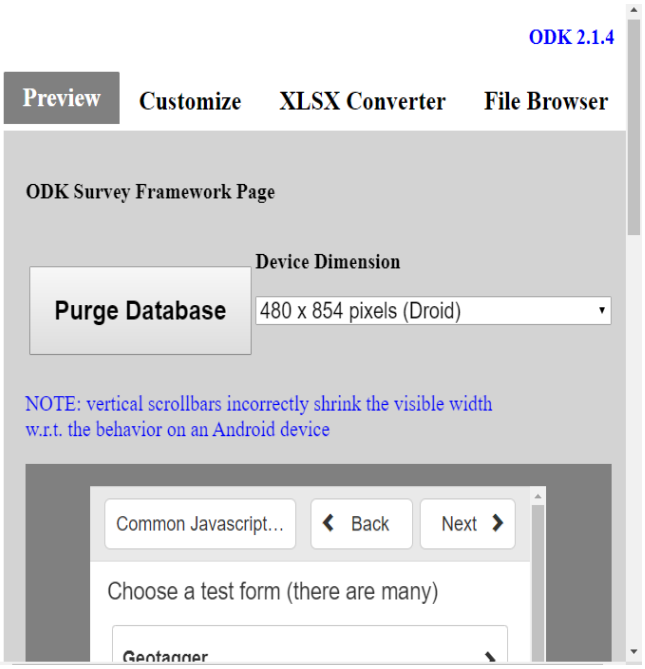


Figure 2.8 : ODK-X - ODK-X Application Designer interface

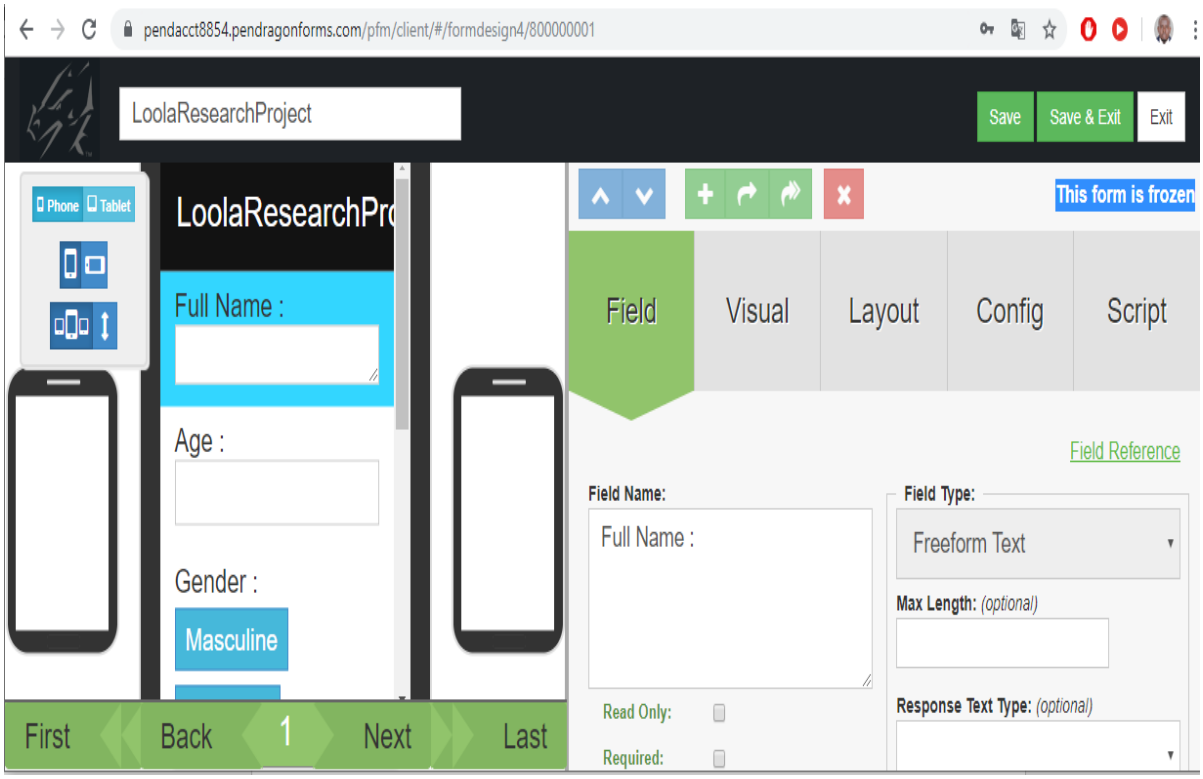


Figure 2.9 : PendragonForms - Form Designer interface

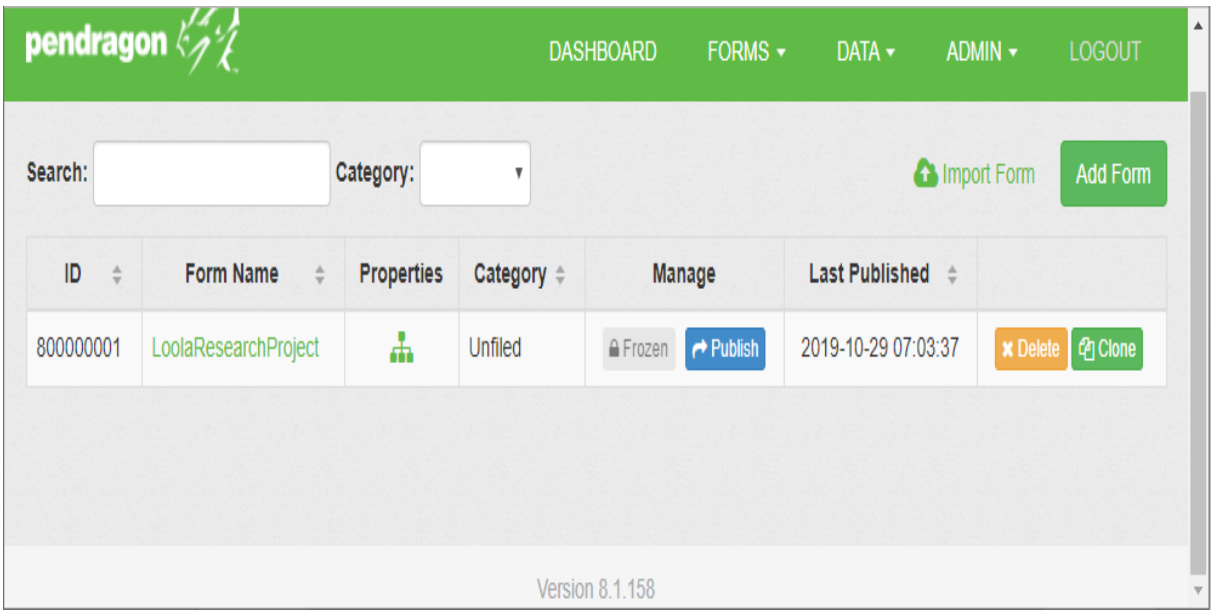


Figure 2.10 : PendragonForms– Server

ODK-X à son tour est assez proche d'ODK à la différence qu'il offre trois applications mobiles dont une, ODK-X Services, doit être obligatoirement installée comme pré requis aux deux autres, à savoir, ODK-X Tables et ODK-X Survey, qui peuvent être utilisées au choix pour la collecte de données. Pour cette étude, nous avons choisi d'utiliser ODK Tables.

Il faudra noter que les applications mobiles d'ODK-X ne se trouvent pas dans la Play Store Google. Pour les installer, il faut télécharger le fichier .apk correspondant à l'application mobile sur le Github du projet puis l'installer sur le mobile. La procédure d'installation est expliquée étape par étape dans la documentation ODK.

2.5. Conclusion

Ce chapitre présente, évalue et compare quatre solutions logicielles de collecte de données par appareils mobiles. Pour aboutir à une comparaison efficace, nous avons choisi d'une part de collecter et sélectionner une documentation sur chacune de ces solutions et d'autre part de tester chacune d'elles en exécutant un processus de collecte de données allant de la création du formulaire jusqu'à la visualisation des données. Cette étude permet d'aboutir aux recommandations ci-dessous.

Pour les entreprises, organisations ou personnes voulant faire de la collecte de données pour une étude ayant un workflow plus ou moins ordinaire, ODK et KoBoToolbox sont très probablement les meilleurs choix. Le choix entre les deux suites sera déterminé par l'autonomie et les compétences techniques de l'équipe qui dirigera l'étude. Si cette équipe a suffisamment de compétences techniques, notamment sur l'installation et la configuration des serveurs, il est préférable d'opter pour ODK, qui a une grande communauté d'utilisateurs et un forum très actif où avoir facilement de l'aide. Si par contre l'équipe n'a pas assez de compétences techniques mais a une forte capacité de compréhension des logiciels (autonomie), dans ce cas, il est préférable d'opter pour KoBoToolbox dont les outils sont très intuitifs et qui n'exige pas d'installer ou de configurer soit même un serveur. Il faudra aussi prendre en compte la communauté d'utilisateurs. ODK est beaucoup plus connu et utilisé que KoBoToolbox, principalement dans le cadre des pays en voie de développement. Il est donc plus facile de trouver des personnes ayant déjà une expérience avec ODK que celle avec KoBoToolbox.

Par ailleurs, pour les entreprises, organisations, personnes voulant faire de la collecte de données pour une étude dont le workflow est complexe et très personnalisé, ODK-X et PendragonForms sont les plus adaptés. Dans ce cas, le choix entre les deux suites se basera sur la capacité financière de l'organisation ainsi que les compétences techniques disponibles.

Si l'entreprise a suffisamment de moyens financiers et peu de compétences techniques en son sein, PendragonForms, qui est une solution commerciale, est la meilleure option. Par contre si l'organisation ne dispose pas d'assez de moyens financiers, ODK-X est le meilleur choix à condition d'avoir au sein de l'équipe des personnes ayant des compétences en programmation.

Il n'y a donc pas une solution logicielle qui soit meilleure en tout point de vue et en toute circonstance. Il faut considérer chaque cas selon le besoin et le contexte des utilisateurs.

Dans le cadre de cette thèse, nous allons opter pour la suite logicielle ODK. Ceci parce que d'une part les enquêtes épidémiologiques, qui sont celles qui nous intéressent, consistent en une collecte de données via de simples questions. Il s'agit là de workflows souvent ordinaires. D'autre part, ODK est parmi toutes les suites, celle qui est la plus répandue et utilisée dans les pays en voie de développement (dont les pays Africains) qui sont notre cible principale. Une autre raison justifiant ce choix est qu'ODK, comme ODK-X, est une plateforme Open Source, nous pouvons donc prendre le code source et en faire ce que nous voulons. Pour finir, lorsque nous considérons les tableaux 2.1 et 2.2 qui donnent une synthèse de comparaison des suites étudiées, ODK a un score bien supérieur à toutes les autres avec sept mentions 'Très bien' sur huit critères gradués, suivi de ODK-X et Pendragon qui en ont chacun cinq, puis KoBoToolbox qui en a quatre. D'où le choix d'utiliser ODK dans le cadre des travaux de cette thèse.

Chapitre 3 : Etat de l'art de la suite logicielle ODK

3.1. Introduction

La collecte, le stockage et l'analyse des données sont des tâches essentielles dans toute organisation. Depuis la nuit des temps, les entreprises, les ONG et les gouvernements ont manifesté le désir d'avoir des outils adaptés pour accomplir ces tâches précitées pour divers besoins. Ces besoins peuvent être : des enquêtes médicales pour améliorer le système de santé, le recensement de la population d'un pays ou des enquêtes d'opinion pour cibler une nouvelle clientèle.

L'utilisation des formulaires en papier fut l'une des premières réponses à cette problématique. Mais la difficulté de rassembler les données venant de milieux géographiquement très éloignés en utilisant les formulaires en papier, par exemple lors d'un recensement dans un pays comme la RDC qui a une superficie de 2 345 000 Km^2 (PopulationData, 2020) , ainsi que la difficulté de conserver sous ce format un nombre important et sans cesse croissant d'informations, en plus des erreurs de transcription, ont démontré les limites de cette solution.

Une invention relativement récente vient sauver la mise : Le Smartphone. Commercialisés pour la première fois vers 2007 (Park & Chen, 2007) , les Smartphones ont pris une place très importante dans la société d'aujourd'hui. Et cela dans les pays développés ou dans ceux qui sont en voie de développement. Cette large expansion des Smartphones est due principalement à leur portabilité, à la facilité d'utilisation, aux choix diversifiés d'applications qu'ils offrent et à leur prix de plus en plus bas.

Ce nouvel outil a inspiré les chercheurs et ingénieurs à remplacer les formulaires en papier par les solutions mobiles pour la collecte, le stockage et l'analyse des données.

Ces dernières années, de plus en plus de chercheurs et ingénieurs ont proposé plusieurs outils de collecte de données par téléphone mobile.

Open Data Kit (ODK) (Anokwa, et al., 2009) est l'une des suites d'outils les plus connues dans ce domaine et beaucoup de chercheurs ont travaillé à le faire évoluer pour qu'il soit très performant afin de mieux répondre aux besoins diversifiés des utilisateurs.

Après une analyse et comparaison de plusieurs suites logicielles mobiles de collecte de données effectuées dans le chapitre 2, ce chapitre se focalise sur Open Data Kit (ODK) et ses extensions.

Ce chapitre répond au besoin d'avoir un état de l'art traçant les efforts des chercheurs sur la suite logicielle Open Data Kit. Le but est de mettre en lumière les contributions qui ont fait évoluer cette suite d'outils tout en donnant des exemples concrets d'utilisation, principalement dans le domaine de la santé. Pour y arriver deux principales questions de recherche ont été posées :

QR1 : Quelles sont les extensions d'ODK proposées dans la littérature ?

QR2 : Quels sont les domaines d'utilisation d'ODK ?

Les réponses à ces questions permettront de présenter une revue de la littérature bien structurée sur Open Data Kit qui servira de référence pour les futurs travaux de recherche en vue d'améliorer l'analyse de données dans la suite logicielle.

Ce chapitre est structuré comme suit : la section 3.2 expose les deux architectures ODK et ODK-X ; la section 3.3 décrit la méthode de recherche suivie pour collecter et sélectionner les articles; la section 3.4 donne le résultat obtenu après l'application de ladite méthode ; la section 3.5 présente une synthèse de l'état de l'art en plus d'une discussion portant sur les articles définitivement retenus; la section 3.6 clôt ce chapitre par la conclusion et les travaux futurs découlant de cette étude.

3.2. ODK et ODK-X

3.2.1. ODK

Open Data Kit (ODK ou ODK 1) est une suite d'outils open source permettant aux organisations de collecter et de gérer leurs données. Nous présentons dans cette sous-section, sur la Figure 3.1, son architecture ainsi que ses outils.

Selon la documentation officielle d'ODK (ODK's Docs, 2019), les principaux outils de cette suite sont au nombre de six. Dans cette étude, pour une meilleure compréhension de l'architecture d'ODK, les six outils ont été regroupés sous trois (3) entités. Les relations entre les éléments des entités sont regroupées en quatre flux qui définissent quatre trajectoires différentes que peuvent prendre les données.

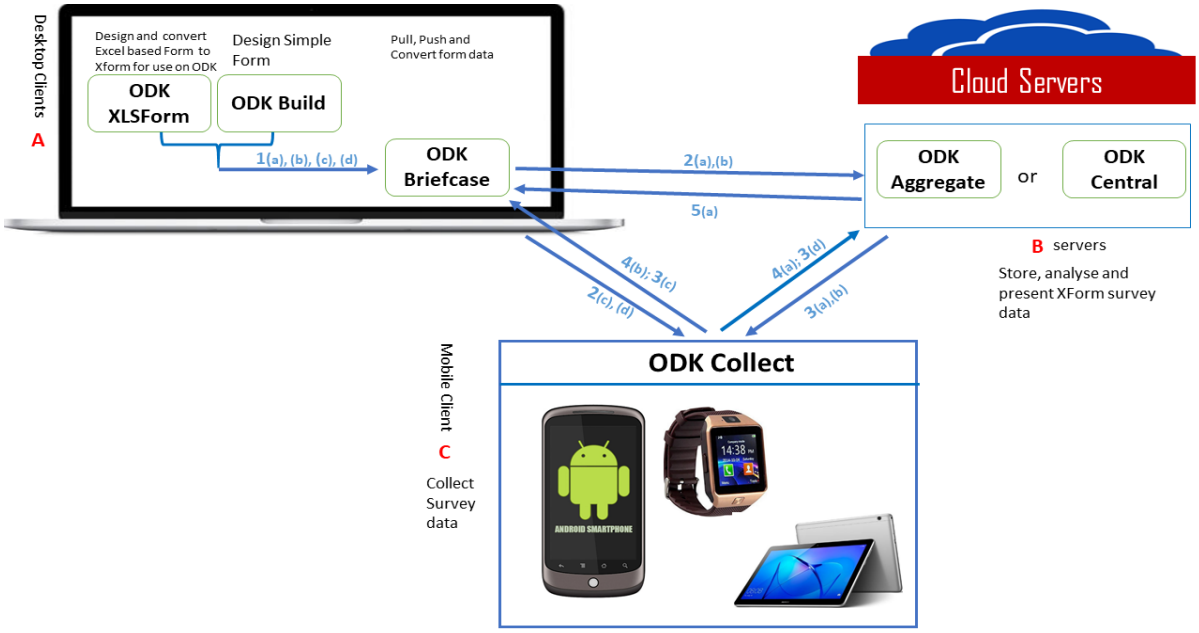


Figure 3.1 : ODK Architecture

- ***Entité A : Clients Desktop***

Les outils de cette entité sont, pour deux d'entre eux, des outils de conception des formulaires :

ODK Build : C'est l'outil d'ODK qui permet de créer les formulaires qui seront ensuite utilisés via ODK Collect. *ODK Build* est une application web qui fonctionne en mode glisser-déposer.

ODK XLSForm : Alors qu'*ODK Build* permet de créer de simples formulaires, *XLSForm* permet la création de formulaires plus complexes. Il permet en effet, de transformer les formulaires créés avec Excel en format Xform pris en charge par les outils d'ODK.

ODK Briefcase : ODK Briefcase est une application de bureau permettant d'extraire, de pousser et de convertir des données de formulaire sur des serveurs ODK tels qu'ODK Aggregate. (ODK's Docs, 2019)

- ***Entité B : Serveurs***

ODK offre deux outils de stockage :

ODK Aggregate : C'est le serveur d'ODK. Il permet le stockage, l'analyse et la visualisation de données collectées grâce à ODK Collect.

ODK Central : C'est une alternative à ODK Aggregate. La particularité d'ODK Central est qu'il prend en charge les nouvelles technologies notamment l'API REST. (ODK's Docs, 2019).

- ***Entité C : Client Mobile***

Cette entité contient le seul outil d'ODK permettant de collecter les données : ***ODK Collect***.

ODK Collect : c'est l'outil d'ODK qui remplace directement les formulaires en papier utilisés autrefois pour collecter les informations.

ODK Collect permet aux utilisateurs, au travers d'une suite d'écrans, de remplir un formulaire en suivant la logique imposée par le concepteur du formulaire et de soumettre le formulaire ainsi rempli. (ODK's Docs, 2019).

- ***Les flux de données***

L'architecture présentée à la Figure 3.1, propose à l'utilisateur quatre trajectoires possibles ((a), (b), (c) et (d)) que peuvent emprunter les données, partant de la création des formulaires jusqu'au stockage de données en passant par la collecte. Nous décrivons dans ce qui suit, les différents flux échangés entre les entités A, B et C.

- 1(a), (b), (c), (d) : le formulaire est créé en utilisant ODK Build ou Excel (puis adapté via ODK XLSForm) et stocké dans le disque dur de l'ordinateur sous le format .xml afin d'être utilisé par ODK Briefcase.
- 2(a), (b) : A l'aide d'ODK Briefcase, le formulaire précédemment créé et sauvegardé est uploadé sur le serveur via le réseau internet.
- 3(a), (b) : A l'aide d'ODK Collect (sur un appareil mobile), on peut télécharger ledit formulaire à partir du serveur.
- 4(a) : Après la collecte de données, le formulaire ainsi rempli est à nouveau uploadé sur le serveur, via le réseau internet.
- 5(a) : Le formulaire est ensuite importé du serveur vers l'ordinateur grâce à ODK Briefcase, via le réseau internet.
- 4(b) : ODK Briefcase donne aussi la possibilité d'importer le formulaire directement à partir d'ODK Collect. Le formulaire pourra par la suite suivre le workflow (a), si nécessaire.
- 2(c), (d) : Le formulaire peut directement être transféré du pc (ordinateur local) vers l'appareil mobile, moyennant un câble ou un réseau sans fil (WiFi, Bluetooth...). Pour cela il faut brancher l'appareil mobile au pc ensuite placer le formulaire (le fichier .xml) dans le dossier odk de l'appareil mobile. Le dossier odk peut soit se trouver dans la mémoire de l'appareil mobile ou sur la carte mémoire.

- 3(c) : Après la collecte de données, on peut à nouveau brancher l'appareil mobile au pc et transférer le formulaire contenant les données sur pc. Le formulaire pourra ensuite suivre la trajectoire (a) ou (b) selon le choix effectué
- 3(d) : A ce stade, il y a aussi la possibilité d'utiliser à nouveau le réseau internet, s'il est disponible pour envoyer les données (formulaire rempli) vers le serveur.

3.2.2. ODK-X

Open Data Kit X (ODK-X) anciennement appelé ODK-2.0 ou ODK-2, désigne la nouvelle suite d'outils d'ODK. Nous présentons dans cette section, l'architecture sur la Figure 3.2 et les outils de cette suite logicielle.

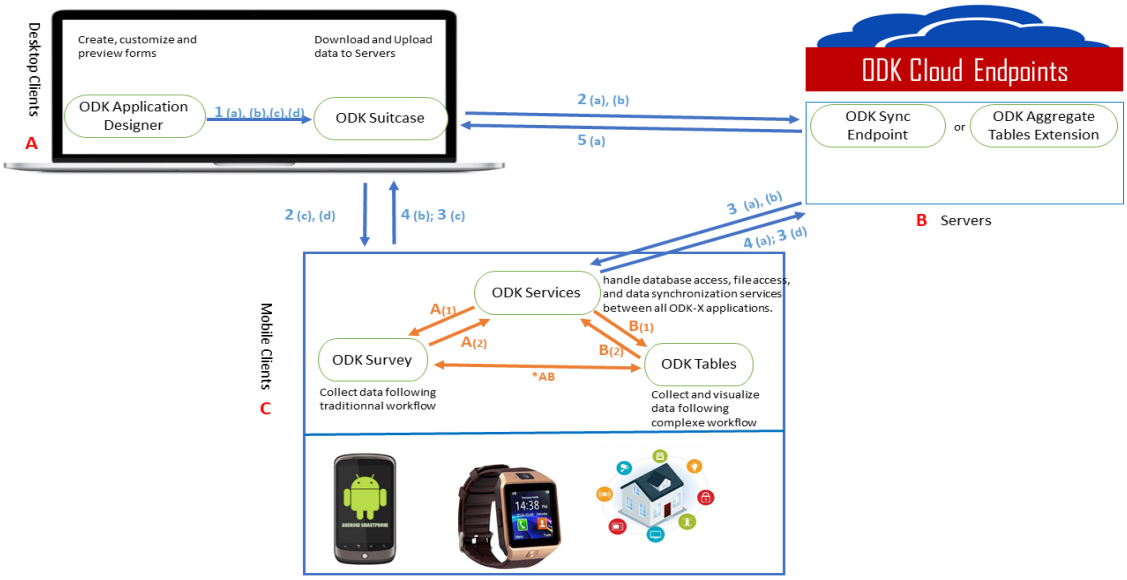


Figure 3.2 : ODK-X Architecture

La documentation officielle d'ODK 2 (ODK-X, 2019) présente sept principaux outils. Dans cette étude, pour une meilleure compréhension de l'architecture d'ODK 2, les sept outils ont été regroupés sous trois (3) entités en différenciant deux flux de données.

- ***Entité A : Clients Desktop***

ODK Application Designer : C'est l'outil qui permet de créer les applications de gestion de données dans ODK-X.

ODK Suitcase : C'est l'outil sur ordinateur qui permet d'accéder aux données stockées dans le serveur ODK 2. (ODK-X, 2019)

- ***Entité C : Client Mobile***

ODK Survey : C'est l'équivalent d'ODK Collect dans ODK 2. Il sert à collecter les données comme son homologue Collect à la différence qu'il est plus flexible dans sa présentation et exécution.

ODK Services : *ODK Services* est installé sur Android comme un prérequis à l'utilisation d'autres outils d'ODK 2. Il permet de faire la synchronisation de données entres toutes les applications ODK 2 et de gérer le service d'accès à la base de données et aux fichiers.

ODK Tables : Permet, sur son portable (Android), de visualiser et de mettre à jour les données existantes. En plus de cela, *ODK Tables,* permet de créer des applications de gestion de données ayant des workflows complexes.

GeoODK ***:*** C'est un outil qui est encore au stade expérimental. C'est une application Android qui permet de récupérer directement les informations sur un formulaire en papier. Avec ODK Scan, l'utilisateur prend en photo le formulaire en papier et l'application s'occupe d'y récupérer les données (ODK-X, 2019).

- ***Entité B : Serveurs***

ODK Cloud Endpoints : Ce sont les serveurs qui communiquent avec les applications ODK 2. A ce jour, on distingue deux Cloud Endpoints pour la communication avec les outils ODK 2 :

ODK Sync Endpoint et ***ODK Aggregate Tables Extension*** qui est simplement ***ODK Aggregate (v1.4.15)*** d'ODK 1 avec une configuration particulière permettant de prendre en charge les outils d'ODK 2 (ODK-X, 2019).

- ***Les flux de données***

Quatre trajectoires possibles sont aussi proposées dans cette architecture d'ODK-X, illustrée à la figure 2, en plus desquelles s'ajoutent deux autres trajectoires que les données peuvent emprunter entre les composants des appareils mobiles. Nous décrivons dans ce qui suit, les flux échangés entre les entités A, B et C.

- 1(a), (b), (c), (d) : le formulaire est créé en utilisant ODK Application Designer et stocké dans le disque dur de l'ordinateur sous le format .csv afin d'être envoyé au serveur par ODK Suitcase.
- 2(a), (b) : A l'aide d'ODK Suitcase le formulaire précédemment créé et sauvegardé est uploadé sur le serveur via le réseau internet.

- 3(a), (b) : ODK Services permet à l'un des deux composants mobiles (Survey et Tables) d'importer le formulaire à partir du serveur.
- 4(a) : Après la collecte de données, le formulaire ainsi rempli est à nouveau uploadé sur le serveur via le réseau internet.
- 5(a) : Le formulaire est ensuite importé du serveur vers l'ordinateur grâce à ODK Suitcase via le réseau internet.
- 4(b) : ODK Suitcase ne donne pas la possibilité d'importer le formulaire directement à partir d'ODK Survey ou ODK Tables sans passer par le serveur donc le recours au mode connecté est nécessaire. Mais cela peut se faire en branchant l'appareil mobile sur le pc (via un câble ou réseau sans fil) puis copier le formulaire sur pc. Le formulaire peut alors suivre la trajectoire (a).
- 2(c), (d) : Via un câble ou réseau sans fil, on peut directement passer un formulaire du pc vers l'appareil mobile. Pour cela il faut brancher l'appareil mobile au pc ensuite placer le formulaire (le fichier. json) dans le dossier odk de l'appareil mobile. Le dossier odk peut soit se trouver dans la mémoire de l'appareil mobile ou sur la carte mémoire.
- 3(c) : Après la collecte de données, on peut à nouveau brancher l'appareil mobile au pc et transférer le formulaire ayant les données sur pc. Le formulaire pourra ensuite suivre la trajectoire (a) ou (b) selon le choix effectué.
- 3(d) : A ce stade, il y a aussi la possibilité d'utiliser à nouveau le réseau internet, s'il est disponible pour envoyer les données (formulaire rempli) vers le serveur.
- A(1) et A(2) : Les données sont collectées en utilisant ODK Survey.
- B(1) et B(2) : Les données sont collectées en utilisant ODK Tables.
- *AB : Montre la possibilité de communication entre ODK Survey et ODK Tables. Les données collectées via ODK Survey peuvent être visualisées et même modifiées avec ODK Tables.

3.3. Méthode de recherche

La méthode de recherche suivie se résume en deux phases, décrite dans Figure 3.3 :

- *Phase 1 :* Identification des moteurs de recherche/bases de données scientifiques à consulter et recherche par mots-clés.
- *Phase 2 :* Sélection des articles.

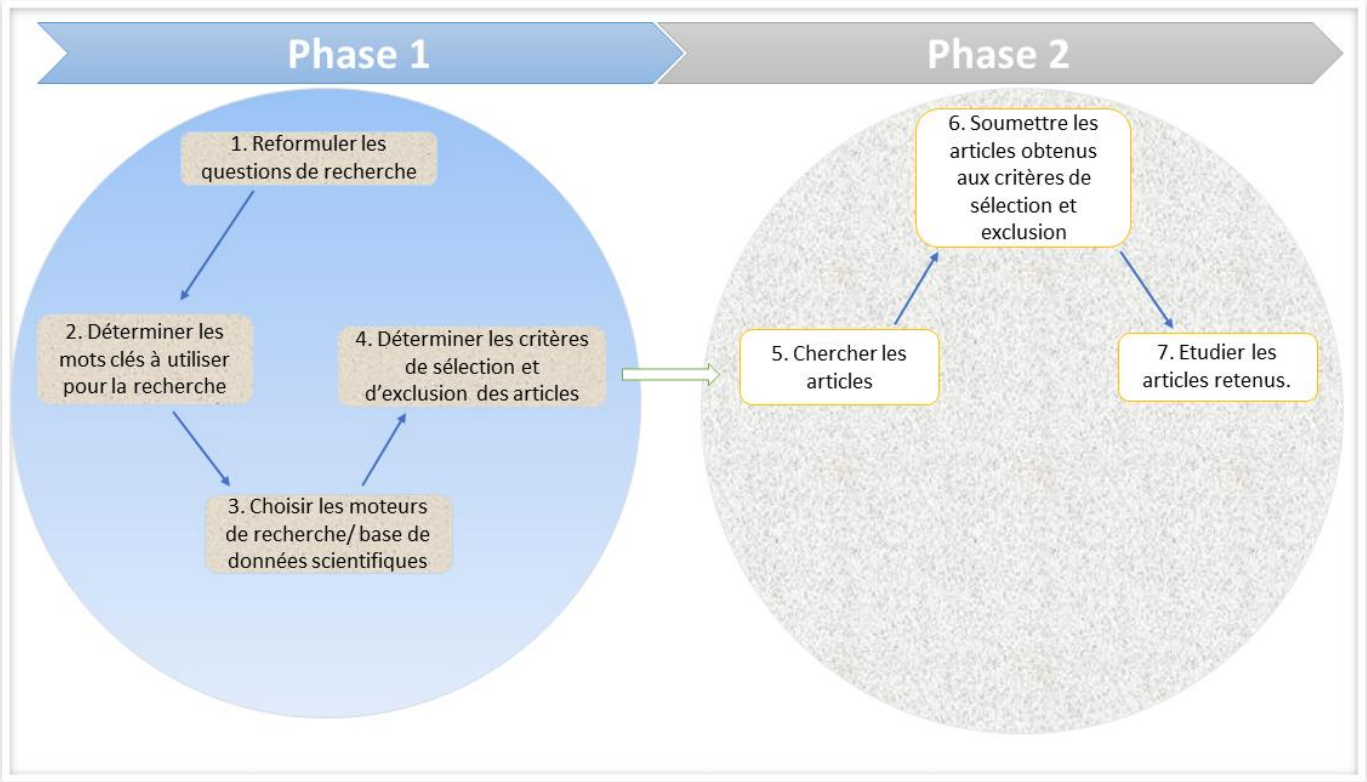


Figure 3.3 : Méthode de Recherche

En janvier 2019, la recherche a été faite dans les moteurs de recherche et bases de données scientifiques suivants : Google Scholar ; Springer Link et Science Direct.

En utilisant des mots clés de recherche spécifiques, notamment : *Open Data Kit (ODK) ; Open Data Kit (ODK) extension ; Open Data Kit (ODK) architecture ; Open Data Kit (ODK) workflow et Open Data Kit (ODK) visualisation.*

Après obtention d'un nombre considérable d'articles, l'élimination et la sélection s'est faite sur la base des critères et moyens suivants, que nous avons résumé sur la Figure 3.4 :

(1) Utilisation des outils de classement par pertinence fournis par les bases de données scientifiques.
(2) Lecture du titre et/ou de l'abstract pour identifier les articles qui répondent à l'une des questions de recherche.
(3) Priorisation des articles publiés après avril 2008 (date de création de l'ODK).
(4) Priorisation des articles publiés dans des revues scientifiques avec comité de lecture.
(5) Critères relatifs à la première question de recherche :
    a. Priorisation des articles traitant de l'extension ou de l'amélioration d'ODK ;
    b. Priorisation des articles traitant d'une extension de l'ODK dans le domaine de la santé.

(6) Critères relatifs à la deuxième question de recherche :

a. Priorisation des articles traitant de l'application de l'ODK ;

b. Priorisation des articles traitant de l'utilisation d'ODK dans le secteur de la santé et dans les pays en voie de développement.

(7) Examen minutieux de l'abstract des articles.

(8) Lecture intégrale du contenu des articles.

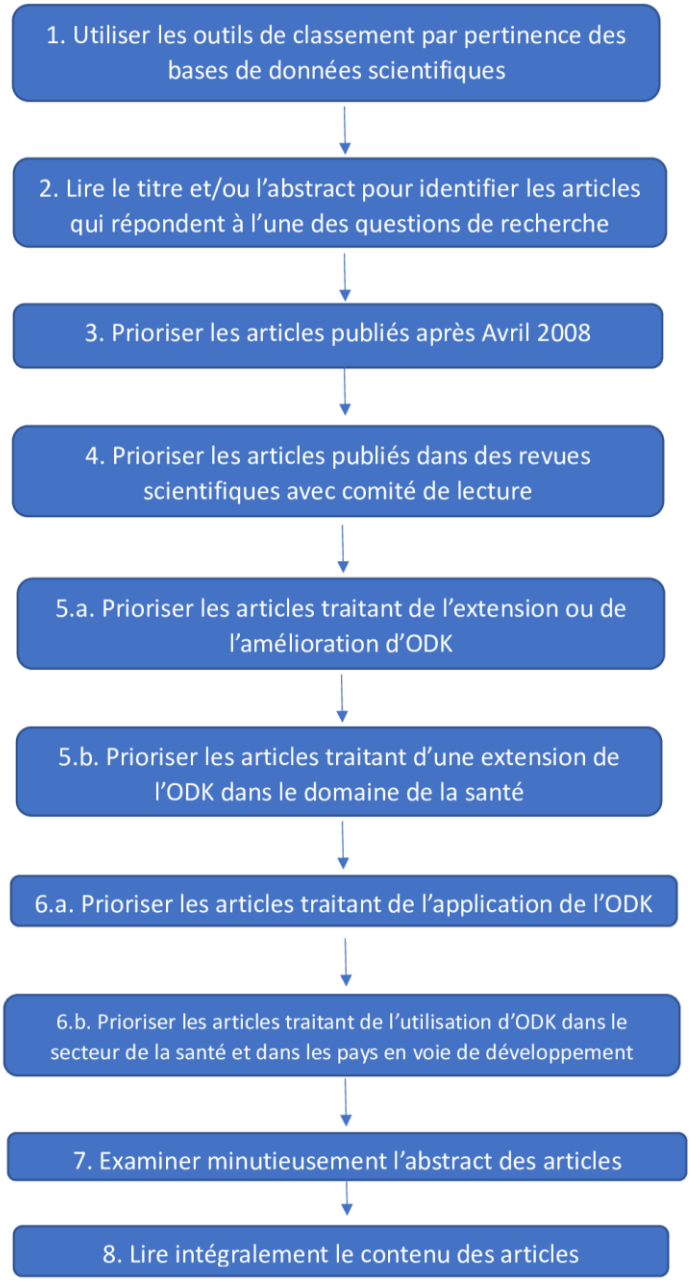


Figure 3.4 : Critères de sélection/exclusion des articles

3.4. Résultats

La recherche effectuée dans les bases de données scientifiques susmentionnées, en utilisant successivement les mots-clés de recherches choisies dans chacune d'elle a donné 4 261 occurrences pour toutes les bases de données confondues. Le Tableau 3.1 montre le nombre d'articles obtenus par base de données et par combinaison de mots - clés.

Tableau 3.1 : Nombre d'articles obtenus par base de données et par combinaison des mots clés

Nombre d'articles obtenus par base de données et par combinaison des mots clés		
Moteur de recherche (Base de données)	**Mots clés**	**Nombre d'articles obtenus**
Google Scholar	Open+Data+Kit+ODK	2380
	Open+Data+Kit(ODK)+extension	572
	Open+Data+Kit(ODK)+architecture	530
	Open+Data+Kit(ODK)+workflow	232
	Open+Data+Kit(ODK)+visualisation	116
	Total Google Scholar :	**3 830**
Springer Link	Open+Data+Kit+ODK	149
	Open+Data+Kit(ODK)+extension	35
	Open+Data+Kit(ODK)+architecture	21
	Open+Data+Kit(ODK)+workflow	6
	Open+Data+Kit(ODK)+visualisation	19
	Total Springer Link:	**230**
Science Direct	Open+Data+Kit+ODK	86
	Open+Data+Kit(ODK)+extension	15
	Open+Data+Kit(ODK)+architecture	22
	Open+Data+Kit(ODK)+workflow	11
	Open+Data+Kit(ODK)+visualisation	6
	Total Science Direct	**140**
TOTAL DES ARTICLES OBTENUS		***4 200***

En appliquant les critères (1), (2) et (3) mentionnés précédemment, nous avons pu éliminer les articles qui ne répondaient pas à nos questions de recherche. Ce premier filtrage nous a permis d'identifier quatre-vingt-douze (92) articles potentiellement intéressants pour notre recherche.

Partant de ces quatre-vingt-douze articles, nous leur avons appliqué les critères (4), (5) et (6), pour extraire ceux qui feront l'objet d'une étude approfondie qui servira de fil conducteur pour les travaux futurs.

En supplément de ces critères, une analyse minutieuse des résumés selon le critère (7) et la lecture intégrale du contenu de chacun de ces quatre-vingt-douze articles selon le critère (8) ont permis d'identifier, trente (30) articles qui ont été adoptés comme objets d'études pour le présent travail. Dans la section suivante sera donné un état de l'art de ces trente-trois articles.

3.5. Synthèse de recherche et discussion

Dans cette section, nous présentons et discutons les travaux de recherche étudiés dans le cadre de cet état de l'art sur ODK et ODK-X (anciennement appelé ODK-2).

3.5.1 Résumé des articles de recherche étudiés.

Après avoir présenté la nécessité qu'il y a dans les pays en voie de développement plus que dans les pays développés d'avoir des applications mobiles de collecte de données, Anoka Y. et al. dans (Anokwa, et al., 2009) expliquent les raisons qui ont poussé au développement d'Open Data Kit malgré l'existence de plusieurs autres systèmes de collecte de données destinés aux régions en voie de développement. La suite de cet article, (Anokwa, et al., 2009), présente les trois premiers outils de la suite Open Data Kit : ODK Collect, ODK Aggregate et ODK Manage tout en donnant plusieurs exemples d'utilisation de l'ODK, entre autres : La surveillance de la déforestation en Amazonie, l'aide à la décision pour les patients pédiatriques en Tanzanie ou un programme de traitement du VIH dans l'ouest du Kenya. L'article met l'accent sur ce dernier cas qui est une mission menée par Ampath pour intensifier son programme de conseil et dépistage à domicile (HCT) au Kenya.

En 2010, Hartung C. et al. (C. Hartung & W. Brunette, 2010) présente le nouveau design de l'architecture de la plateforme ODK en ce temps-là et ses quatre principaux outils : ODK Collect, ODK Aggregate, ODK Voice et ODK Build. On y trouve une comparaison entre ODK et d'autres plateformes de collecte de données telles que : CyberTracker, historiquement la plus ancienne dans ce domaine et Pendragon une solution payante mais aussi avec les plateformes

open source telles que : Front-lineForms, EpiSurveyor, CommCare, et JavaRosa qui sont des solutions J2ME (Java Platform Micro Edition). La méthode suivie dans cet article était de choisir quatre organisations qui ont utilisé ODK et ont donné des avis objectifs et clairs tout en expliquant pourquoi ODK était meilleur ou pas par rapport à ces concurrents. Ces organisations sont: Berkeley Human Rights Center, D-Tree International, Johns Hopkins Center for Clinical Global Health Education et Academic Model for the Prevention and Treatment of HIV.

Le but principal de l'article de Hong Y. et al. (Hong Hilary, et al., 2011) est de présenter ODK Tables, un composant de la suite Open Data Kit 2, nouvellement créé. Les auteurs expliquent qu'ODK Tables permet de collecter et d'organiser les données dans des tableaux (tables) d'une base de données hébergée sur un téléphone mobile (SQLite databases). Il s'agit d'une organisation des données, comme on le ferait avec Excel ou Access sur les ordinateurs. ODK Table fournit aussi la possibilité de parcourir, réorganiser, indexer et modifier les tableaux créés. La communication par SMS est aussi prise en charge par ODK Tables. Il est possible d'envoyer des requêtes aux tableaux ou d'insérer une nouvelle ligne via la messagerie SMS.

En outre, ODK Tables permet d'importer un fichier aux formats .csv ou .xls ce qui permet la création automatique d'un tableau ayant le contenu du fichier importé. L'export d'un fichier sous le format .csv ou .xls est aussi possible. ODK Tables est connecté à ODK Collecte et ODK Aggregate afin que les données soient soumises aux serveurs pour assurer la synchronisation avec d'autres téléphones. L'article donne des cas d'utilisation d'ODK Tables dans six domaines différents, notamment : la collecte automatique de données, la détection participative, la microfinance, les recherches basiques, la recherche des prix du marché et la planification. On y trouve aussi une brève comparaison avec des composants des plateformes concurrentes qui proposent un service presque similaire, il s'agit de : FrontlineSMS, RapidSMS, et UjU.

Chaudhri R. propose ODK Sensors dans (R. Chaudhri, et al., 2012), une extension d'Open Data Kit qui a pour but de faciliter l'intégration des capteurs au sein des applications mobiles de collecte de données. Du point de vue des utilisateurs, ODK Sensors se compose de deux applications : *Application App* et *Framework App*, le premier se charge de communiquer avec des capteurs via une API alors que le deuxième gère les communications de bas niveau et spécifiques aux canaux et fournit des abstractions permettant d'isoler le code du pilote du capteur.

La suite de l'article décrit le Framework ODK Sensors. Pour montrer l'efficacité du Framework, l'article présente trois applications qui ont été développées en se servant du Framework ODK

Sensors. Ces trois applications sont des exemples de trois catégories de capteurs de collecte de données. L'article présente aussi une comparaison entre ODK Sensors et d'autres Frameworks tels que : Reflex, Zhuang, PRISM, IOIO ou Phidgets.

Dans (W. Brunette, et al., 2012), Brunette W. et al. parlent de ODK Sensors comme un Framework qui facilite l'interaction entre une application Android et plusieurs capteurs externes utilisant des canaux de communication et formats de données différents. Après avoir montré le lien étroit qu'il y a entre Open Data Kit (ODK) et ODK Sensors, les auteurs précisent qu'ODK Sensors est destiné à trois types de personnes ayant des compétences à différentes échelles, notamment : *les utilisateurs d'application ; les développeurs d'application et les développeurs des pilotes des capteurs.* Les auteurs donnent un fonctionnement détaillé du Framework. Pour une meilleure compréhension, les auteurs ont choisi de développer trois versions du Framework : *dans la v1 la communication avec les pilotes du capteur se fait via des méthodes internes au Framework ; dans la v2 cette communication se fait via les IPCs à partir d'une interface de service et dans la v3 elle se fait par broadcast.*

Cet article de Brunette et al. (Brunette, et al., 2013) a pour vocation de présenter la nouvelle architecture d'ODK qui est ODK 2.0. Les auteurs expliquent que cette nouvelle architecture vient répondre aux préoccupations exprimées par les utilisateurs d'ODK. Les améliorations apportées à ODK couvrent quatre domaines : *la prise en charge des fonctions d'agrégation, d'analyse et de visualisation de données directement sur le téléphone en permettant à l'utilisateur de visualiser et modifier les données à partir de son téléphone ; l'augmentation de la possibilité de personnaliser une application pour qu'elle soit plus adaptée au besoin ; élargissement des types de données qui peuvent être collectées à partir des appareils de détection.*

Après avoir présenté les limitations d'ODK 1, l'article présente le nouveau design de la plateforme à savoir ODK 2.0. Dans la suite une brève comparaison est faite avec d'autres solutions existantes telles que : CAM ; MyExperience ; Commcare ; Uju ; Reflex. La conclusion de cette comparaison est qu'ODK 2.0 se distingue des autres solutions par l'interopérabilité de ces éléments et la possibilité d'effectuer toutes les opérations sur téléphone.

Plus tard, vers la fin de l'année 2013 Brunette et al. présentent dans (Brunette, et al., 2013) ODK Survey, un composant d'ODK 2.0. Les auteurs expliquent, qu'après avoir remarqué qu'ODK n'était pas suffisamment flexible pour répondre aux besoins de plus en plus divers des utilisateurs, une enquête a été menée pour comprendre la raison. Il en est sorti que plusieurs

organisations avaient des difficultés à développer des applications adaptées à leur workflow en utilisant les interfaces utilisateurs génériques d'ODK et JavaRosa XForm.

C'est pour répondre à ce besoin qu'ODK Survey a été développé. ODK Survey améliore la capacité d'adaptation d'ODK aux workflows des organisations dans la mesure où il permet d'implémenter des workflows non linéaires. Avec ODK Survey l'ordre des informations collectées est spécifié par l'utilisateur. Dans la suite, l'article présente en détail le fonctionnement d'ODK Survey avant de donner dans la section 3 un exemple de cas d'utilisation qui implémente le workflow de l'Organisation Mondiale de la Santé (OMS) pour la gestion intégrée des maladies infantiles, personnalisé pour un déploiement au Ghana et en Inde.

Gupta A. et al. présentent (Gupta, et al., 2013) une extension d'ODK. Ladite extension porte le nom d'ODK-Ext. Le travail réalisé consistait à ajouter deux nouvelles fonctionnalités à ODK-Collect. La première est l'ajout du concept des formulaires pré-remplis à l'aide des données préalablement collectées. La deuxième est l'ajout de la fonctionnalité de validation des SMS. Après avoir expliqué l'utilité de ces deux nouvelles fonctionnalités, l'article présente l'architecture d'ODK-Ext. ODK-Ext a été utilisé dans deux états différents de l'inde. Les résultats préliminaires de ces deux déploiements sont présentés dans la section 4 de l'article. Il s'agit de : Andhra Pradesh Survey et Bihar Survey.

Macharia P. et al. à leurs tours (Macharia, et al., 2013) se sont donnés pour but d'étudier si l'utilisation des applications mobiles dans le secteur de la santé (m-Health) peut être bénéfique, pour les pays en voie de développement. Ce besoin est motivé d'une part par le manque de ressources et infrastructures appropriées dans les pays en voie de développement, rendant ainsi la collecte de données difficile et d'autre part par le besoin des décideurs et professionnels de la santé d'avoir des données précises pour pouvoir évaluer l'efficacité des politiques et des programmes existants dans le système de santé et en définir de nouveaux.

C'est dans cet objectif que l'équipe TLC-IDU a configuré ODK pour évaluer sa facilité d'utilisation en tant que solution M-Health. Cette étude réalisée par le Programme national de lutte contre le SIDA et les IST (NASCOP) du ministère de la Santé du gouvernement du Kenya et l'Université de New York aux États-Unis d'Amérique a été faite dans dix (10) sites dont quatre (4) à Nairobi et six (6) à Mombasa auprès de 1947 participants.

Les résultats satisfaisants d'ODK ont permis d'aboutir à la conclusion selon laquelle l'utilisation des M-Health peut améliorer la productivité des agents de santé mobiles et améliorer la qualité, le coût et l'efficacité des soins.

En 2015, Kipf A. et al. (Kipf, et al., 2016) ont proposé une extension d'ODK 2.0 appelé Mezuri. La plateforme proposée (Mezuri) utilise l'infrastructure d'ODK 2.0 et d'autres systèmes de collecte de données des capteurs tels que Get All The Data et SWEETSense. Mezuri est une plateforme complète de collecte, gestion et diffusion de données, conçue pour être utilisée par les managers des programmes de développement mondial et les chercheurs.

Dans l'article, les auteurs font une comparaison avec les autres plateformes existantes telles que : Kepler, Conveyor, Ta- verna, Mobyle, DHIS2 et Open Foris. Les auteurs donnent les limitations de ces plateformes précitées, dont le nombre de langages de programmation supportés pour certaines d'entre elles ou la limitation à un domaine spécifique pour d'autres. L'article donne par ailleurs quatre exemples d'application de la nouvelle plateforme. Mezuri est une plateforme intégrée de collecte de données avec provenance, traitement, analyse et partage.

Cette étude (Brunette, et al., 2015) présente un nouvel outil ajouté à ODK 2.0. Il s'agit d'ODK Submit. Le but d'ODK Submit est d'optimiser la transmission de données dans les milieux où la connexion internet est difficile. Ce but est atteint en utilisant plusieurs connexions différentes (ex. : cellular, Wi-Fi, peer-to-peer) en se basant sur les propriétés des données et la connexion disponible.

Les auteurs commencent par explorer et caractériser diverses méthodes de transmission de données utilisant : *les outils de synchronisation existants ; la communication peer-to-peer ; et les réseaux rares (sparse networks).*

Cet article (Brunette, et al., 2017) parle de la possibilité qu'offre ODK 2.0 aux utilisateurs de créer et personnaliser les applications mobiles de gestion de données, selon leur niveau de compétences techniques. L'article présente les limitations d'ODK 1.x ainsi que les recommandations qui ont poussé à la création d'ODK 2.0.

On y trouve aussi une présentation détaillée des six composants d'ODK 2.0, il s'agit de : *Services ; Survey ; Tables ; Scan ; Sensors et Submit.* Les auteurs présentent en outre six études de cas d'ODK 2.0 impliquant plusieurs organisations. Ces études sont: *1) Childhood*

Pneumonia; 2) Chimpanzee Behavior Tracking; 3) HIV Clinical Trial; 4) Disaster Response; 5) Mosquito Infection Tracking; et 6) Tuberculosis Patient Records.

La vocation principale de cet article, (Brunette, 2017), est de présenter le travail de Waylon Brunette. On y découvre que les travaux de recherche de Waylon Brunette se focalisent sur la création d'outils permettant d'adapter les technologies mobiles aux besoins et contexte des populations des pays en voie de développement. L'auteur explique que malgré que la grande majorité de la population mondiale possède des téléphones portables (plus de deux tiers de la population mondiale), moins de la moitié de ladite population est connectée sur internet. C'est ce constat qui poussa Waylon Brunette et d'autres chercheurs à créer Open Data Kit (ODK), une plateforme mobile qui permet de collecter les données avec un téléphone déconnecté d'internet.

Dans la suite de l'article, l'auteur explique deux points principaux qui différencient ODK des autres plateformes mobiles de collecte de données : *1) ODK permet à une organisation ou une personne n'ayant pas de fortes connaissances en programmation de créer et personnaliser une application, 2) ODK peut être utilisé dans un environnement sans connexion internet.* L'auteur finit son article en montrant la progression de la plateforme ODK.

Cet article, (Chaudhri, et al., 2013)**,** présente une amélioration apportée à ODK Sensors. Cette amélioration consiste à rendre ODK Sensors capable de communiquer avec des capteurs de bas niveau ou utilisant des interfaces numériques analogiques. Ceci a permis d'augmenter la diversité des applications mobiles de détection.

Les auteurs expliquent que dans la version précédente d'ODK Sensors, le canal USB du framework n'était connecté qu'à un pont USB basé sur Arduino via le kit de développement d'accessoires Android. Mais depuis lors, les téléphones Android ont introduit des modes de communication USB. ODK Sensors a donc été adapté pour prendre en charge ces nouveaux modes.

Cet article, (Brunette, et al., 2013), traite d'une implémentation particulière d'ODK Tables. Le projet ICTD (Information and Communication Technologies and Development) a effectué un ensemble de cas d'utilisation auprès des organisations utilisant la suite ODK, dans les pays en voie de développement. Ces cas d'utilisation, dont un sous-ensemble de cinq (5) est présenté dans l'article, ont permis de déduire neuf (9) besoins des utilisateurs qui n'étaient pas suffisamment comblés par la suite ODK. Le but de cette implémentation est d'apporter une

solution à cinq (5) d'entre eux qui sont : - *la mise à jour de données existantes et les requêtes personnalisées pour extraire des informations utiles ; - le partage et la synchronisation de données entre plusieurs mobiles ; - l'affichage de données de différentes manières ; - Plus de flexibilité pour être adapté à différents besoins ; - l'extensibilité à d'autres serveurs.* L'article s'achève par une évaluation préliminaire comprenant des chiffres basiques de performance ainsi qu'un aperçu des travaux futurs.

Cet article, (Signore, 2016), traite de l'utilisation d'Open Data Kit dans le domaine d'agro biodiversité. Les auteurs ont utilisé ODK en combinaison avec Google Fusion Tables pour collecter et partager les données relatives à l'agro biodiversité. En l'occurrence ODK collect a été utilisé pour collecter les informations concernant les cultures végétales à risque d'érosion alors que Google Fusion Tables a servi à partager ces informations avec un grand public.

L'utilisation de ces outils a été faite en temps réel dans la région de Puglia en Italie et le résultat fut satisfaisant. L'auteur conclu son article en disant qu'ODK est un outil simple et fiable pour collecter des informations sur l'agro-biodiversité.

Cette étude, (JeffreyCoker, et al., 2010), a été menée au Mali par les chercheurs de l'université Columbia de New York. Durant cette étude, plus de 300 agriculteurs maliens ont été interrogés sur les statistiques démographiques familiales, leurs histoires personnelles, et les pratiques agricoles. Les enquêtes sur terrain ont été menées par les travailleurs de l'entreprise Malienne MBSA, une société de production de biodiesel qui travaille directement avec les agriculteurs. ODK a été utilisé comme outil de collecte de données, ODK Collect, et de stockage, ODK Aggregate.

Outre les limitations et problèmes rencontrés tels que : le maintien de la charge de la batterie à long terme ; difficultés de transmission de données liées à la connexion internet pour lesquelles les auteurs proposent *Kobo Postprocessor* comme un outil complémentaire à ODK permettant la synchronisation de données hors ligne ; prise en charge des caractères spéciaux et des lettres accentuées (en Français) dont la solution est d'utiliser le code html correspondant au caractère voulu.

L'étude se conclut par l'affirmation disant que l'utilisation d'ODK comme outil de collecte de données est bénéfique à la fois pour les projets locaux et internationaux.

Cet article, (Giduthuri, et al., 2014), décrit une étude portant sur l'acceptation du vaccin antigrippal, qui a eu lieu à Pune, une ville de l'Inde. L'enquête a été menée à la fois avec les

formulaires en papier et les formulaires ODK. L'utilisation parallèle de ces deux méthodes a permis d'obtenir les résultats et conclusions suivants : *Le taux d'erreur des entretiens avec les formulaires en papier est relativement identique à celui des entretiens avec les formulaires ODK (2,01% et 1,99% respectivement). Les enquêteurs qui ont manifesté une préférence entre les deux méthodes ont opté pour les formulaires ODK. Pour de petites études, le coût d'investissement pour l'utilisation des formulaires ODK est supérieur à celui nécessaire à l'utilisation des formulaires en papier ; mais ce coût devient inférieur pour de grandes études.*

Cet article, (Tom-Aba,, et al., 2015)**,** traite d'une étude menée au Nigeria ayant pour but de doter les membres de l'équipe d'intervention de la lutte contre l'épidémie d'Ebola des technologies et solutions permettant d'obtenir un flux de données fluide et rapide. Deux principales technologies ont été utilisées : Open Data Kit (ODK) et FormHub. FormHub jouait le rôle du serveur pendant que ODK a permis de collecter les informations telles que : *la présence ou l'absence des symptômes du Virus Ebola (VE), la température ainsi que les coordonnées GPS du lieu de la visite*. Une fois les données soumises au serveur, si la personne interrogée présente des symptômes du VE, une alarme est immédiatement déclenchée sur le serveur et un message est envoyé aux responsables du programme afin de mettre en place un plan d'évacuation. Le résultat final a démontré qu'il y a eu une nette amélioration du service après le déploiement et l'utilisation de ces technologies.

Cet article, (Bell, et al., 2016) **,** présente les premiers résultats d'une étude utilisant un modèle de microtask pour micropaiements afin de collecter un ensemble de données sur les niveaux de vie dans les communautés et les ménages dans une zone rurale de Bangladesh. Les enquêteurs ont utilisé des téléphones avec une interface personnalisée d'ODK pour collecter l'information. Les participants à l'étude répondaient à de petits questionnaires de trois à dix minutes en retour d'un paiement sous forme de données mobiles prépayées, de messagerie SMS et de temps de conversation. C'est ce que les auteurs appellent 'Microtasks for Mircropayments'. Dans cette zone rurale et difficilement accessible, l'utilisation des téléphones mobiles Android dotés d'ODK a largement facilité la collecte rapide de l'information en temps réel, permettant ainsi aux décideurs d'avoir des données significatives pour intervenir efficacement.

Cet article, (Jain, et al., 2018)**,** présente l'utilisation ainsi que l'apport de GeoODK dans le processus de la cartographie, la numérisation et la visualisation des routes. GeoODK est l'acronyme de Geographical Open Data Kit. C'est l'extension d'ODK qui offre la possibilité de collecter et de stocker les informations géo référencées. Pour couvrir le processus précité, l'architecture proposée est la suivante : la cartographie (le mapping) est gérée du côté mobile

(GeoODK Collect), le stockage et la visualisation du côté serveur *(ODK Aggregate),* la numérisation du côté client (le navigateur). Comme cas d'utilisation, le réseau routier de Dehradun, une zone d'étude située au Nord de l'Inde et Capitale de l'Etat de l'Uttarakhand a été développée.

Fort du constat que les soins à domicile sont la manière la plus efficace pour apporter les soins de santé aux populations des pays en voie de développement, les auteurs de (Rajput, et al., 2012) proposent dans cet article un système basé sur les téléphones mobiles qui pourra guider les agents de santé communautaires dans le processus de dépistage de la population afin de détecter la présence du VIH et d'autres risques de santé.

Ce système a été implémenté en utilisant ODK Collect. Son efficacité a été testée et approuvée lors d'un programme dirigé par l'USAID en partenariat avec AMPATH dans l'Ouest du Kenya. Ledit programme avait pour but d'organiser des visites à domicile auprès de deux millions d'individus. Les utilisateurs du système ont estimé qu'il était facile à utiliser et facilitait leur travail. Le système était également plus rentable que les alternatives au stylo et au papier.

Cet article, (Medhanyie, et al., 2015), décrit une étude qui a été réalisée en Ethiopie auprès des sages-femmes et des aides-soignants. Cette étude consistait à évaluer les expériences, les obstacles, les préférences et facteurs de motivation des agents de santé lors de l'utilisation des formulaires médicaux sur des smartphones dans le contexte des soins de santé maternelle en Ethiopie, dans le but d'introduire l'utilisation de tels formulaires dans les établissements de soins de santé primaires.

Les formulaires ont été développés en utilisant ODK avec quelques configurations dont la prise en charge de la langue locale et du calendrier local. A l'issue de l'expérimentation, les sages-femmes et les aides-soignants ont trouvé les formulaires électroniques très utiles pour la prestation quotidienne de leurs services de soins de santé maternelle.

Cet article, (Tridane, et al., 2014), présente le prototype d'un outil (nommé REACap) conçu pour traiter la collecte de données sur la grippe saisonnière. Plusieurs technologies et plateformes ont contribué à mettre en place cet outil. Mais la plus grande part revient à ODK. Les architectes de cet outil ont choisi d'utiliser ODK Build et ODK Collect pour respectivement la création des formulaires et la collecte des données. REACap fait partie intégrante du projet AREA dont l'objectif est de faire de la recherche sur l'utilisation des applications mobiles pour la surveillance de la santé, l'analyse et la prévision de données.

Cette étude, (Fornace, et al., 2018) , a eu lieu dans le but d'améliorer l'identification des schémas spatiaux des maladies à petite échelle, dans les zones à faibles ressources et sans adresse officielles. Les auteurs évaluent l'utilisation des applications Android contenant la fonctionnalité de géolocalisation. Les fonctionnalités, la convivialité et le coût des trois logiciels conçus pour collecter des informations spatiales ont été comparés : GeoODK, ArcGIS-Survey123 et ePAL. Il en est sorti que malgré que les trois logiciels présentent presque les mêmes avantages à quelques différences près, GeoODK a le plus gros avantage d'être totalement gratuit, contrairement aux deux autres.

Cet article, (Jeffers, et al., 2019) , décrit une étude qui a eu lieu dans huit villages de la région de Velondriake à l'Ouest de Madagascar. C'est dans cette région que se trouve la plus grande concentration des pécheurs traditionnels des requins. L'économie de la région est fortement dépendante de cette pêche. Le but de l'étude est d'évaluer le potentiel des smartphones et de la plateforme Open Data Kit à aider à la collection de données sur la pêche des requins. Elle fait suite à une étude précédente réalisée dans la même région en utilisant les papiers pour collecter les données. Ce qui a permis de comparer la vitesse, la précision et l'expérience utilisateur de deux méthodes. Il en est sorti que les smartphones (et donc ODK) constituent un outil utile et précis pour la collecte participative de données sur la pêche. Cependant, cela convient mieux aux régions à couverture mobile plus étendue.

Cet article, (Pavluck, et al., 2014), parle du développement d'un système mobile de collecte de données permettant de cibler d'importants défis en matière de santé publique dans une région. Pour développer ce système, plusieurs plateformes mobiles de collecte de données ont été étudiées dont *EpiCollect, FormHub, EpiInfo et ODK*. La restriction de la licence et bien d'autres défis ont fait en sorte que les autres plateformes soient mises de côté au profit d'ODK. Le système LINKS construit à partir d'ODK a été utilisé dans plusieurs pays tels que : Ethiopie, Tanzanie, Kenya, Mozambique, Nigeria, République Dominicaine, et Inde.

Cet article, (Shiferaw, et al., 2018), décrit le processus de développement et les aspects techniques d'un système de santé mobile (mHealth). Ce système a pour but d'aider les agents de santé de niveau intermédiaire à fournir de meilleurs services de soins de santé maternelle en automatisant la collecte de données et le processus de prise de décision. Pour l'implémentation de ce système, trois composants d'ODK ont été considérés : *ODK Build, ODK Collect et ODK Aggregate*. L'étude a été menée en Ethiopie au nord de la zone de Shewa dans la région d'Amhara.

Le but de cette étude, (Little, et al., 2013), est de répondre aux besoins techniques des agents de vulgarisation sanitaire et de sages-femmes pour la santé maternelle à l'aide d'outils et de technologies mobiles appropriés. Pour ce faire, plusieurs plateformes mobiles ont été étudiées, suite à cette étude ODK fut retenu. Les raisons qui ont poussé au choix d'ODK sont : a) *Open Source : Donc une grande facilité de personnalisation ; b) Prend en charge la norme XForms : Prend donc en charge une grande variété de types de questions (texte, numérique, choix multiples, etc), en plus peut gérer le système de positionnement global (GPS), information de localisation, photos, vidéos, audio et codes à barres ; c) la flexibilité de s'étendre à d'autres domaines, tels que le contrôle de base de stocks et la vaccination.*

Cette étude, (Ginsburg, et al., 2015), porte sur le développement d'une application nommée « mPneumonia ». mPneumonia est une application Android de santé publique dont le but est d'aider à réduire la mortalité par pneumonie chez les enfants et améliorer la capacité des fournisseurs de soins de santé primaire à diagnostiquer, classer et gérer la pneumonie et d'autres maladies de l'enfant. Le développement de « mPneumonia » a eu lieu grâce à la collaboration entre PATH et l'université de Washington. mPneumonia utilise ODK Survey et toute la suite ODK 2.0 pour implémenter le protocole IMCI (Integrated Management of Childhood Illness) de l'OMS qui était autrefois appliqué sur des papiers.

3.5.2. Avantages d'ODK sur les autres plateformes concurrentes

Le résultat de cette revue de littérature a mis en lumière trois principales caractéristiques qui différencient la plateforme ODK des autres. La première d'elles est sa gratuité (Anokwa, et al., 2009). Bien qu'il existe d'autres solutions de collecte de données sur mobiles, certaines d'entre elles sont payantes à l'exemple de Pendragon (Pendragon, 2021), qui est une solution commerciale et complète. L'aspect open source d'ODK a énormément facilité son adoption dans les pays en voie de développement où le pouvoir d'achat de la population est assez faible.

En outre, parmi les solutions open source, ODK marque la différence par l'ajout en 2011 d'ODK Tables (Hong Hilary, et al., 2011), un outil qui permet la collecte de données par SMS. Ce qui enlève l'obligation d'avoir des téléphones sophistiqués pour effectuer une collecte électronique de données. Cette fonctionnalité de validation par sms a été ensuite ajoutée en 2013 à ODK Collect (Gupta, et al., 2013).

La troisième mais pas la moindre des particularités d'ODK est la possibilité de soumettre les données même lorsque le téléphone n'est pas connecté à internet. L'ajout d'ODK Submit (Brunette, et al., 2015) a permis à la plateforme de mieux s'adapter au contexte des pays en

voie de développement où la connexion internet n'est pas toujours au rendez-vous. ODK Submit permet aux utilisateurs d'ODK de soumettre leurs données au serveur même lorsqu'ils sont dans un milieu sans connexion internet.

3.5.3. Comment faire un choix entre ODK et ODK-X ?

La documentation officielle d'ODK 2 (ODK-X, 2019), actuellement nommé ODK-X, suggère de commencer par utiliser ODK 1.x et de passer à ODK 2 uniquement si ODK 1.x ne répond pas convenablement au besoin.

Dans la même documentation, les auteurs précisent qu'ODK 2 a été créé pour coexister avec ODK 1 et non pour le remplacer. Il serait donc une erreur de penser qu'ODK 2 est forcément meilleur.

Pour des personnes ou organisations n'ayant pas des compétences techniques en programmation et dont le workflow n'est pas très complexe, il est conseillé de choisir ODK 1 qui est facile d'utilisation.

En revanche, si la logique d'implémentation de l'étude est très complexe et que l'utilisateur a besoin d'un outil très flexible, ODK 2 sera le meilleur choix. Néanmoins, il est à noter que quelques compétences techniques sont nécessaires pour prendre en main ODK 2.

3.6. Conclusion

Ce chapitre s'intéresse à la collecte de données par appareil mobile via ODK. ODK est une plateforme mobile de collecte de données qui se distingue des autres plateformes parce qu'elle offre une solution complète et totalement open source et aussi par son objectif principal qui est de mettre en place une suite d'outils interopérables qui sont personnalisables par un non-programmeur selon le contexte de déploiement et qui peut fonctionner dans des environnements déconnectés.

ODK et ODK-X ont fait un bout de chemin ensemble avant de se séparer complétement. Au début, la même équipe qui a développé ODK a aussi créé ODK-X qu'elle a d'abord baptisé ODK 2.0. Les mois et années passés, ODK 2.0 a été rebaptisé en ODK-X toujours avec la même équipe. A ce jour, les deux suites comportent chacune son équipe de développement et maintenance et évoluent séparément. Aucune d'elles n'a vocation à remplacer l'autre, bien au contraire, elles sont complémentaires, elles répondent à des besoins différents.

ODK qui reste jusqu'à ce jour, la suite la plus populaire des deux, est destinée à une utilisation pour des workflows qui ne sont pas très complexes et ne demandent pas une forte

personnalisation. Alors qu'ODK-X répond au besoin inverse. Les deux plateformes sont Open Source. Elles peuvent donc faire l'objet d'une modification de code source par qui le souhaite.

Dans le cadre de cette thèse, comme mentionné dans le chapitre 2, nous optons pour ODK au vu des raisons citées dans ledit chapitre. Dans la suite de ce mémoire, nous nous intéresserons aux méthodes d'analyse prédictive avant de revenir sur ODK pour une expérimentation finale.

Chapitre 4 : Etat de l'Art des Méthodes de Machine Learning pour l'Analyse Prédictive Epidémiologique

4.1. Introduction

Depuis une dizaine d'années, l'Intelligence Artificielle (IA) connait une renaissance, et particulièrement le Machine Learning (ML) fait l'objet d'une grande attention. Le but ultime de l'IA est de rendre les machines capables de réaliser mieux que l'homme des tâches jusqu'ici considérées comme intelligentes (Université de Montréal, 2016) et (Pigenet, 2017). N'est-ce pas là, la promesse d'une nouvelle révolution qui bouleversera notre rapport au travail et à la connaissance (Pigenet, 2017) ?

Pendant que certains voient en cela une opportunité inespérée d'améliorer nos performances dans plusieurs domaines, d'autres y voient le danger d'un adversaire prêt à prendre la place de l'homme dans divers secteurs. Mais qu'à cela ne tienne, chercheurs, hommes politiques, entreprises et gouvernements tous utilisent l'IA pour diverses raisons : lutte contre le cancer, profiling des électeurs, moteurs de recherche, publicités programmatiques, armement des soldats, météo, traitement de données, etc. (Mosavi, et al., 2018).

Un élément commun à la plupart des cas d'utilisation de l'IA est la prédiction. La prédiction du risque élevé de contracter un cancer (Barlow, et al., 2019), la prédiction par profiling des électeurs les plus susceptibles de voter pour tel ou tel autre candidat, (Newman & Sheth, 1985), la prédiction de vidéos et/ou publicités pouvant intéresser telle ou telle autre personne, etc. (McDuff, et al., 2014). L'analyse prédictive est de plus en plus utilisée.

Mais une question répétitive se pose à chaque fois qu'il faut effectuer une analyse prédictive : Quelle méthode utiliser ? Dans le cadre de cette thèse, nous nous sommes posé la même question. Ce chapitre vient répondre à cette problématique. Nous y réalisons une revue des tendances et des méthodes de ML utilisées pour effectuer une analyse prédictive au sein des études les plus récentes.

Le ML étant une discipline qui a sa propre terminologie, et afin d'habituer les chercheurs nouvellement intéressés par le ML aux concepts propres à cette discipline, nous avons consacré la section 2 à la définition des termes utilisés en ML. Le but étant de faciliter la compréhension, ainsi que l'analyse et des résultats de cette revue.

La démarche entreprise et expliquée dans la section 3 a permis de ne retenir et étudier que les travaux de recherche effectués lors des 3 dernières années : 2020, 2019 et 2018. Le but étant de fournir aux chercheurs, entreprises ou quiconque désirant effectuer une analyse prédictive des

indices lui permettant de choisir la (les) meilleure (s) méthode (s) de ML selon son domaine d'application, en se basant sur les derniers travaux de recherche récents dans la littérature. Nous n'avons retenu dans cette revue que les articles des journaux avec comité de lecture.

Le résultat de cette revue de littérature est présenté dans la section 4. Il s'agit des disciplines où les méthodes de ML sont fréquemment utilisées pour faire l'analyse prédictive, des techniques d'apprentissage les plus utilisées, des méthodes de ML les plus utilisées et même des méthodes de ML les plus utilisées par discipline. Ce résultat ainsi présenté permet non pas de dicter quelle méthode choisir mais de savoir quelles sont celles que les autres chercheurs dans le monde utilisent dans une discipline précise pour faire de l'analyse prédictive. La conclusion de ce chapitre est présentée dans la section 5.

4.2. Définitions et Taxonomie

Force est de constater que la plupart du temps les auteurs utilisent les concepts du Machine Learning sans les définir au préalable. Par exemple, des 30 articles retenus dans le cadre de cette revue de littérature, seuls trois articles donnent clairement une définition d'un ou plusieurs des termes utilisés : (Saint-Cirgue, 2019) (Lima & Delen, 2019) (Kaur & Kumari, 2018).

Le Machine Learning a été initié par Arthur Samuel, un mathématicien américain, en 1959 lorsqu'il a développé un programme capable d'apprendre tout seul comment jouer aux Dames. Arthur Samuel définit le Machine Learning comme étant la science qui consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul, c'est-à-dire le programmer explicitement. (Saint-Cirgue, 2019) (Lima & Delen, 2019) le définissent comme étant : « *Un sous-ensemble de l'intelligence artificielle, qui est souvent appliquée lorsque des dispositifs de calcul tentent d'imiter les fonctions cognitives humaines liées aux processus d'apprentissage et de résolution de problèmes afin d'obtenir des résultats "optimaux"* ». Pour leur part, (Kaur & Kumari, 2018) définissent le Machine Learning comme étant : *« le développement d'algorithmes et de techniques qui permettent aux ordinateurs d'apprendre et d'acquérir une intelligence basée sur l'expérience passée. Il s'agit d'une branche de l'intelligence artificielle (IA) et est étroitement liée aux statistiques. L'apprentissage signifie que le système est capable d'identifier et de comprendre les données entrées, afin de pouvoir prendre des décisions et faire des prédictions sur la base de celles-ci ».* (Moujahid, et al., 2018) définissent le Machine Learning comme étant : *« La discipline qui permet à une machine*

d'améliorer ses performances en tirant les leçons des événements précédents, de la même manière que les humains tirent les leçons de leurs expériences ».

Dans le cadre de cette thèse, nous retenons la définition synthèse suivante : le Machine Learning consiste à former des systèmes capables de comprendre les données entrées afin de prédire des réponses ou d'y extraire des informations utiles. C'est un sous ensemble de l'intelligence artificielle et est étroitement lié aux statistiques.

S'agissant des types d'apprentissage que le Machine Learning (ML) comprend, nous avons constaté que les auteurs ne parlent pas d'une seule voix sur ce sujet. Certains comme (Castañón, 2019) ou (Kaur & Kumari, 2018) distinguent deux types d'apprentissage, *supervisé (supervised) et non supervisé (unsupervised)*. D'autres comme (Paul, et al., 2020) considèrent trois types d'apprentissage, *supervisé, non supervisé et semi-supervisé (semi-supervised)*. (Patel & Jhaveri, 2015) considèrent l'apprentissage par renforcement comme étant le troisième type en supprimant le semi-supervisé de la liste. (Moujahid, et al., 2018) distinguent quatre types d'apprentissage, *supervisé, non supervisé, par renforcement et en profondeur (deep learning)*.

Dans le but d'inclure le maximum de techniques d'apprentissage dans notre étude nous considérons, dans le cadre de cette thèse, quatre types d'apprentissage : supervisé, non supervisé, semi supervisé et par renforcement.

L'apprentissage supervisé est défini par (Patel & Jhaveri, 2015) comme étant : *« Une sorte de technique d'étiquetage, qui a une entrée prédéfinie et une sortie connue générée avec les paramètres du système ».* Sur ce même sujet, les auteurs, (Kaur & Kumari, 2018) disent : *« La technique d'apprentissage supervisé est utilisée lorsque les données historiques sont disponibles pour un certain problème. Le système est formé avec les entrées et les réponses respectives, puis utilisé pour la prédiction des réponses à de nouvelles données ».* Ajoutons l'observation de (Helin, et al., 2020) avant de proposer notre propre définition, *« Dans l'apprentissage supervisé, un ensemble de caractéristiques ou de données est utilisé pour construire le modèle d'apprentissage. Le modèle utilise ces données pour apprendre la relation entre les entrées et les sorties »*. Nous en déduisons la définition synthèse suivante : L'apprentissage supervisé consiste à former un système en lui donnant un ensemble d'entrées et leurs réponses respectives, pour un type de problèmes bien précis. Ensuite, pour le même type de problèmes, lorsqu'une nouvelle entrée est saisie, le système se sert du modèle d'apprentissage préalablement construit pour prédire la réponse correspondante.

L'apprentissage supervisé est subdivisé en deux sous types : classification et régression (Helin, et al., 2020).

La classification consiste à trouver une relation entre des entrées et des sorties discrètes. Les variables de sortie sont également appelées catégories ou étiquettes. Une fonction de cartographie (classificateur) est construite en analysant les données de formation d'entrée dans l'étape d'apprentissage, et ce classificateur est adoptée pour prédire les étiquettes de classe catégorielles dans l'étape de classification (Helin, et al., 2020). La régression quant à elle, consiste à estimer ou à prédire des quantités continues. La régression s'appuie sur des caractéristiques statistiques d'entrée pour établir la relation entre deux ou plusieurs variables indépendantes (Helin, et al., 2020).

L'apprentissage non supervisé, « *est un apprentissage sans étiquette, c'est-à-dire, pas de vecteur de sortie* » (Patel & Jhaveri, 2015). Autrement dit, « *L'apprentissage non supervisé se caractérise par l'absence d'une connaissance exacte de ce que contiennent les données ou de l'objectif final* » (Moujahid, et al., 2018), ou encore *« Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé n'est pas fourni avec des étiquettes (pas de vecteurs de sortie). L'objectif de l'apprentissage non supervisé est d'analyser la structure des données et d'y extraire les informations utiles sans aucune indication explicite du résultat escompté* » (Helin, et al., 2020). L'apprentissage non supervisé peut donc se définir comme étant un apprentissage à l'aveugle, ayant pour principe d'apprendre à partir des entrées saisies. Il consiste à donner au système des entrées sans modèle d'apprentissage préalable afin que ce dernier tire de celles-ci toutes les informations utiles.

L'apprentissage non supervisé comprend deux sous types : le regroupement (clustering) et la réduction de la dimensionnalité (dimensionality reduction) (Helin, et al., 2020).

Le regroupement consiste à diviser un ensemble d'objets en différents groupes de sorte que les objets de chaque groupe soient aussi similaires que possible les uns aux autres, et que les différents groupes soient aussi différents que possible les uns des autres (Helin, et al., 2020). Alors que la réduction de la dimensionnalité vise à transformer un grand espace de données en un espace plus petit sans perdre les informations utiles des données d'origine (Helin, et al., 2020).

L'apprentissage semi-supervisé, *« comme son nom l'indique, est un hybride des deux approches susmentionnées. L'apprentissage semi-supervisé est couramment utilisé lorsque certains cas ont des valeurs pour les covariables (entrées) et les résultats (sortie), mais la majorité des cas n'ont des valeurs que pour les covariables et manquent de données sur le résultat voulu »* (Paul, et al., 2020). *« Dans l'apprentissage semi-supervisé, peu de données étiquetées et de nombreuses données non étiquetées sont utilisées »* (Abbas, et al., 2020). Nous pouvons donc définir l'apprentissage semi-supervisé comme étant un type d'apprentissage dans lequel le système est formé de telle sorte à pouvoir résoudre à la fois les problèmes dont il possède déjà des échantillons d'entrées accompagnées de leurs réponses, et les problèmes dont il n'a aucune idée sur les types de réponses escomptées.

L'apprentissage par renforcement, *« fait référence à un type d'apprentissage machine où l'agent prend des décisions sur les actions à entreprendre dans un certain environnement, afin de maximiser certaines notions de la récompense cumulative »* (Patel & Jhaveri, 2015) *«L'apprentissage par renforcement est un domaine particulier du Machine Learning qui est basé sur le fait d'entreprendre certaines actions, suivies de récompenses numériques pour atteindre un objectif. Le point important est que celui qui entreprend une action, appelé agent, dans un monde particulier, appelé environnement, ne sait pas quelle action est bonne ou mauvaise, mais il apprendra quelles sont celles qui donneront les plus grandes récompenses en les essayant »* (Moujahid, et al., 2018). La définition synthèse que nous retenons pour l'apprentissage par renforcement est la suivante : C'est une technique qui consiste pour un agent à apprendre à agir dans un environnement grâce aux actions qui lui rapportent des récompenses dans le but d'atteindre un objectif.

Pour bien comprendre l'apprentissage par renforcement qui est très différent des trois autres, prenons l'exemple d'un des plus célèbres exploits que l'intelligence artificielle (IA) a réalisé dans cette décennie. Il s'agit d'AlphaGo Zero (AlphaGo, 2017). En effet, l'agent ici, un joueur virtuel, créé grâce à l'apprentissage par renforcement a pu gagner les 60 (soixante) meilleurs joueurs professionnels du monde au jeu AlphaGo (AlphaGo, 2017). L'agent (joueur virtuel) a été créé en utilisant les réseaux de neurones sans connaissances préalable des règles du jeu, au fur et à mesure qu'il avançait dans le jeu, lorsqu'il posait une action qui lui rapportait des points (récompenses), celle-ci était enregistrée comme bonne, ainsi de suite il a appris toutes les règles.

S'agissant des termes tels que : technique, méthode, modèle, et algorithme, nous adopterons le sens et définition proposés par nos pairs. Le mot ***technique*** sera utilisé pour désigner un type d'apprentissage (supervisé, non-supervisé, semi-supervisé et de renforcement) (Kaur & Kumari, 2018), (Stephen, et al., 2019), (Armando J, et al., 2018). Quant à ***méthode,*** il sera utilisé pour désigner les différentes méthodes du Machine Learning *ANN, SVM, DT, etc.* (MBML, 2013). Nous retiendrons la nuance entre algorithme et modèle proposé par (MBML, 2013). Un ***modèle*** est un ensemble d'hypothèses sur un domaine de problème, exprimées sous une forme mathématique précise, qui est utilisé pour créer une solution d'apprentissage automatique (MBML, 2013). Alors qu'un ***algorithme*** est simplement une série d'instructions utilisées pour implémenter un modèle afin de résoudre un problème ou effectuer un calcul.

4.3. Méthode de recherche

La méthode de recherche suivie se résume en deux phases, décrites à la *Figure 4.1*.

- **Phase 1 :** Composée de quatre étapes dont le but principal est d'apprêter et rassembler les éléments nécessaires pour la recherche, la sélection et l'exclusion des articles :

  1. Formuler les questions de recherche ;
  2. Déterminer les mots clés à utiliser pour la recherche ;
  3. Choisir les bases de données scientifiques à utiliser ;
  4. Déterminer les critères d'inclusion et d'exclusion des articles.

- **Phase 2 :** Composée de trois étapes qui utilisent les éléments de la phase 1 :

  5. Chercher les articles dans les bases de données sélectionnées ;
  6. Soumettre les articles obtenus aux critères d'inclusion et d'exclusion ;
  7. Etudier les articles retenus.

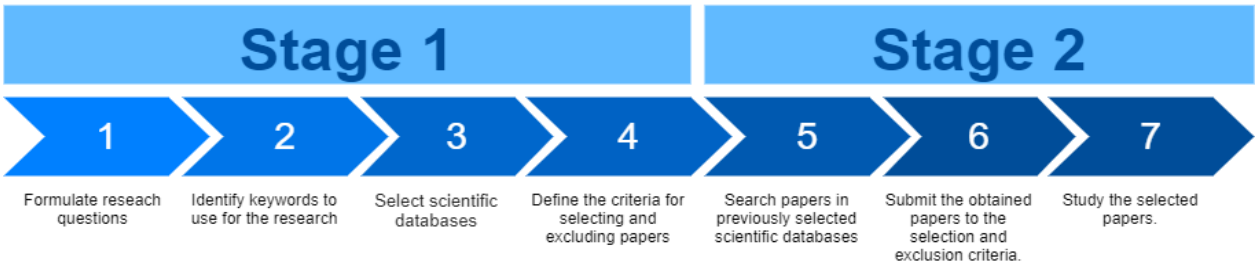


Figure 4.1 : Méthodologie de recherche

Deux questions de recherche (QR) ont guidé cette revue de littérature : (QR1) Quelles sont les méthodes d'analyse prédictive des données proposées dans la littérature ? (QR2) Parmi les articles résultant de la QR1, quels sont ceux qui ont appliqué les méthodes d'analyse prédictive sur des données collectées dans le cadre d'une enquête ? Nous avons utilisé les deux combinaisons de mots - clés suivantes :

- Data+"predictive analysis"+method;
- Data+ "predictive analysis"+ "machine learning"+method.

La recherche effective a eu lieu dans trois bases de données scientifiques notamment *Science Direct, Springer Link et IEEE Xplore.*

Afin de pouvoir répondre aux QR susmentionnées, nous avons considéré les critères suivants :

1. Retenir les articles qui ont été publiés à partir de 2015 ;
2. Priorisation des articles publiés dans des revues scientifiques avec comité de lecture ;
3. Si le nombre d'articles est supérieur à mille (1000), retenir uniquement les articles publiés à partir de 2018 ;
4. Si le nombre est toujours supérieur à mille (1000), retenir uniquement les articles publiés à partir de 2019 ;
5. Utilisation des outils de classement par pertinence fournis par les bases de données scientifiques ;
6. Lecture du titre et/ou de l'abstract pour identifier les articles répondant à la QR1 ;
7. Lecture intégrale des articles pour identifier et retenir les articles répondant à la QR2.

4.4. Résultat de la recherche

4.4.1. Répartition des articles par base de données

La recherche effectuée au sein des trois bases de données scientifiques susmentionnées, en utilisant successivement les mots - clés choisis dans chacune d'elles, a donné 10 095 articles pour toutes les bases de données confondues. Le Tableau 4.1 ci-après illustre la première phase de la recherche :

Tableau 4.1 : Nombre d'articles obtenus par base de données et par combinaisons de mots clés

Base de données	**Mots clés**	**Nombre d'articles obtenus**
Science Direct	Data+"predictive analysis"+method	5 027
	Data+"predictive analysis"+"machine learning"+method	634
	Total Science Direct	**5 661**
Springer Link	Data+"predictive analysis"+method	3 406
	Data+"predictive analysis"+"machine learning"+method	947
	Total Springer Link	**4 353**
IEEE Xplore	Data+"predictive analysis"+method	65
	Data+"predictive analysis"+"machine learning"+method	16
	Total IEEE	**81**
Total		10 095

Après l'application des critères 1) à 6) susmentionnés, nous avons retenus 246 articles. Le Tableau 4.2 donne le nombre d'articles retenus par base de données, suivant les critères 1), 2), 3) 4), 5) et 6) avant la suppression des doublons.

Tableau 4.2 : Nombre d'articles retenus par base de données et pour les combinaisons de mots clés.

Base de données	**Nombre d'articles obtenus entre 2015-2020**	**Nombre d'articles des journaux entre 2018-2020**	**Nombre d'articles des journaux entre 2019-2020**	**Nombre d'articles retenus**
Science Direct	2 573	1448	917	157
Springer Link	1 126	703		72
IEEE Xplore	43			17
Total				246

De ces 246 articles, 13 doublons ont été détectés et supprimés. Les 233 articles restants ont été lus intégralement afin d'identifier ceux répondant à la QR2 (critère 7). A l'issue de cette lecture, 30 articles ont été inclus dans cette revue de littérature.

4.4.2. Répartition des publications par année et par domaine d'application

Dans cette revue de littérature, la méthode de recherche suivie nous a permis de retenir des articles récents et pertinents pour notre revue de littérature. Parmi les 30 articles inclus dans cette revue, six ont été publiés en 2020, vingt et un en 2019 et trois en 2018.

Au fil des années, les chercheurs ont utilisé les méthodes d'analyse prédictive pour résoudre des problèmes de société. Par exemple, la proposition de (José Rômulo, 2020) a permis d'améliorer la distribution des financements des logements destinés aux populations à basses revenues dans le cadre du programme (Yassir, 2020) lancé par le gouvernement brésilien pour plus de 84 milliards de dollars. Celle de Dae-Hee Han (Dae-Hee, 2019) a permis de prédire la mauvaise utilisation des opioïdes chez les adolescents aux USA.

Outre le domaine social, les méthodes d'analyse prédictive ont été appliquées dans plusieurs autres domaines. C'est ainsi que le travail de Camila Maione (Camila, 2019) a consisté à prédire l'origine florale et géographique du miel alors que celui de Jongko Choi (Jongko, 2018) a pour but de développer un modèle de prédiction qui identifie le risque potentiel d'accident mortel dans les chantiers de construction en Corée du Sud.

Le domaine de l'éducation a été exploré entre autres par (Ann & Yahia Mohamed, 2020) en Egypte avec le développement d'une plateforme pour applications mobiles afin de prédire la performance académique de futurs étudiants ; le même travail a été effectué pour les établissements publics au Brésil (Shahid Mohammad & Majid, 2021).

Le domaine médical a fait l'objet de plusieurs études dont celle d'Amir Talaei-Khoei (Amir, 2018) qui permet d'identifier les patients ayant un risque de développer plusieurs complications médicales dues à une maladie chronique, ou celle de (Beaulah Mary, et al., 2019) dont l'objectif est de prédire l'efficacité d'un traitement anti cancer afin de donner à chaque patient un traitement adéquat, ou alors celle de (Kaur & Kumari, 2018) dont l'objet est la détection précoce du diabète chez les femmes.

En tout, les articles retenus couvrent cinq domaines d'application : la médecine pour 56,7% des articles, les sciences sociales pour 26,7%, l'éducation pour 10%, la botanique pour 3,3%, et la construction pour 3,3%.

Comme mentionné ci-haut, 56% des travaux de recherche inclus dans cette revue ont pour domaine d'application la médecine. La médecine étant un domaine très vaste, nous avons

énumérer cinq spécialités notamment : la cardiologie (Abbas, et al., 2020), (Tova M., et al., 2020) ; l'oncologie (Lin, et al., 2019), (Beaulah Mary, et al., 2019), (Gu-Wei, et al., 2019); la diabétologie (Stephen, et al., 2019), (Kaur & Kumari, 2018), (Hang, et al., 2019) ; la psychiatrie (Lee, et al., 2018) et la chirurgie pédiatrique (Armando J, et al., 2018), en plus desquelles il faudra ajouter la médecine générale qui regroupe le plus grand nombre d'articles (Bumberger, et al., 2019), (Araz, et al., 2019), (Abbas, et al., 2020) , (Lee, et al., 2018), (Li, et al., 2019), (Brian L., et al., 2019), (Shahid Mohammad & Majid, 2021). Nous pouvons déjà noter qu'aucun article n'a utilisé une maladie épidémiologique.

Le Tableau 4.3 liste les articles inclus dans cette revue, leurs années de publication ainsi que leurs domaines d'application. La Figure 4.2 donne un aperçu des domaines d'application et la Figure 4.3 donne un aperçu général des années de publication des articles étudiés.

Tableau 4.3 : Articles par domaine et année de publication

Article	**Domaine**	**Année**
Predicting adults likely to develop heart failure using readily available clinical information (Tova M., et al., 2020)	Médecine Cardiologie	2020
Using machine learning to predict opioid misuse among U.S. adolescents (Shahid Mohammad & Majid, 2021)	Sciences Sociales	
Machine learning models for credit analysis improvements: Predicting low-income families' default (José Rômulo, et al., 2019)	Sciences Sociales	
A Mobile Application for Early Prediction of Student Performance Using Fuzzy Logic and Artificial Neural Networks (Ann & Yahia Mohamed, 2020)	Education	
Machine learning predictive model based on national data for fatal accidents of construction workers (Jongko, et al., 2020)	Construction	
Testing the convergent- and predictive validity of a multi-dimensional belief-based scale for attitude towards personal safety on public bus/ minibus for long-distance trips in Ghana: A SEM analysis (José Rômulo, et al., 2019)	Sciences Sociales	
Predictors of length of stay in the coronary care unit in patient with acute coronary syndrome based on data mining methods (Abbas, et al., 2020)	Médecine Cardiologie	
Application of the Albumin-Bilirubin Grade in Predicting the Prognosis of Patients With Hepatocellular Carcinoma: A Systematic Review and Meta-Analysis (Lin, et al., 2019)	Médecine Oncologie	
Computational models for predicting anticancer drug efficacy: A multi linear regression analysis based on molecular, cellular and clinical data of oral squamous cell carcinoma cohort (Beaulah Mary, et al., 2019)	Médecine Oncologie	

Machine-learning analysis of contrast-enhanced CT radiomics predicts recurrence of hepatocellular carcinoma after resection: A multi-institutional study (Gu-Wei, et al., 2019)	Médecine Oncologie	2019
Predicting the botanical and geographical origin of honey with multivariate data analysis and machine learning techniques: A review (Camila, et al., 2019)	Botanique	
Can we predict lesion detection rates in second-look ultrasound of MRI detected breast lesions? A systematic analysis (Bumberger, et al., 2019)	Médecine Générale	
Predictive analytics for hospital admissions from the emergency department using triage information (Araz, et al., 2019)	Médecine Générale	
A predictive analytics framework for identifying patients at risk of developing multiple medical complications caused by chronic diseases (Talaei-Khoei, et al., 2019)	Médecine Générale	
Using Machine Learning Applied to Real-World Healthcare Data for Predictive Analytics: An Applied Example in Bariatric Surgery (Stephen, et al., 2019)	Médecine Diabétologie	
Predictive Analytics and Modeling Employing Machine Learning Technology: The Next Step in Data Sharing, Analysis, and Individualized Counseling Explored With a Large, Prospective Prenatal Hydronephrosis Database (Armando J, et al., 2018)	Médecine Chirurgie pédiatrique	
Residential demand response program: Predictive analytics, virtual storage model and its optimization (Saurav, et al., 2019)	Sciences Sociales	
Predicting and explaining corruption across countries: A machine learning Approach (Lima & Delen, 2019)	Sciences Sociales	
Utilizing early engagement and machine learning to predict student outcomes (Gray & Perkins, 2019)	Education	
Identifying predictors of probable posttraumatic stress disorder in children and adolescents with earthquake exposure: A longitudinal study using a machine learning approach (Fenfen, et al., 2020)	Sciences Sociales	
Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records (Li, et al., 2019)	Médecine Générale	
An automated machine learning-based model predicts postoperative mortality using readily-extractable preoperative electronic health record data (Brian L., et al., 2019)	Médecine Générale	
Educational data mining: Predictive analysis of academic performance of public-school students in the capital of Brazil (Fernandes, et al., 2019)	Education	
Ensemble method based predictive model for analyzing disease datasets: a predictive analysis approach (Dharavath & Yogendra, 2019)	Médecine Générale	
A Predictive Analytics-Based Decision Support System for Drug Courts (M.Zolbanin & Dursu, 2018)	Sciences Sociales	
Preventing Infant Maltreatment with Predictive Analytics: Applying Ethical Principles to Evidence Based Child Welfare Policy (Paul, et al., 2020)	Sciences Sociales	

Predictive models for diabetes mellitus using machine learning techniques (Hang, et al., 2019)	Médecine Diabetologie	
Predictive modelling and analytics for diabetes using a machine learning Approach (Kaur & Kumari, 2018)	Médecine Diabetologie	2018
Processing electronic medical records to improve predictive analytics outcomes for hospital readmissions (M.Zolbanin & Dursu, 2018)	Médecine Générale	
Applications of machine learning algorithms to predict therapeutic outcomes in depression: A meta-analysis and systematic review (Lee, et al., 2018)	Médecine Psychiatrie	

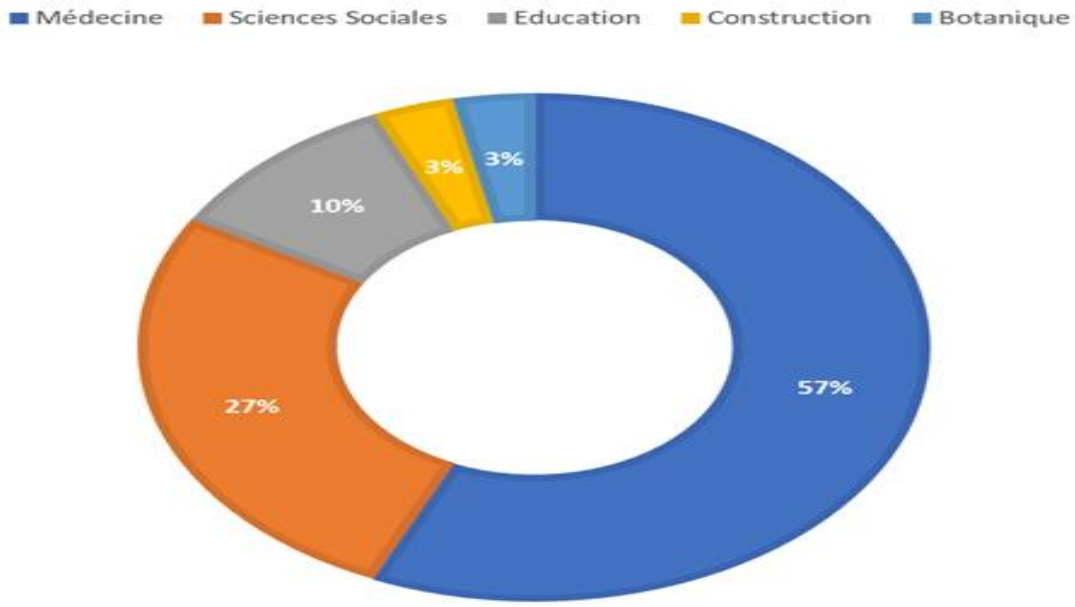


Figure 4.2 : Domaines d'application des articles étudiés

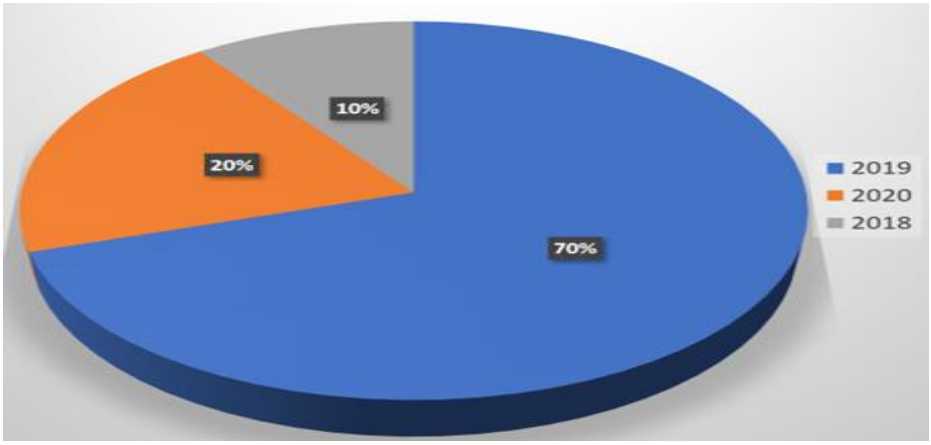


Figure 4.3 : Années de publication des articles étudiés

4.5. Méthodes ML utilisées dans les articles étudiés

Nous pouvons différencier entre deux grands groupes de méthodes d'analyse prédictive. Les méthodes basées sur l'utilisation des statistiques de base et les méthodes de ML parfois appelées méthodes de datamining.

En comparant ces deux approches (Mojtaba, et al., 2018), qui ont utilisé les quatre méthodes ML suivantes : ANN, SVM, DT, AC, sont arrivés à cette conclusion : *« L'utilisation de modèles classiques pour expliquer les prédicteurs de résultats est inefficace lorsque le nombre de prédicteurs est élevé et la taille de l'échantillon faible. Par conséquent, l'analyse basée sur de nouvelles approches de datamining est une solution alternative souhaitable. »*

En se basant sur cette conclusion de (Mojtaba, et al., 2018), dans la suite de cette revue de littérature, nous nous intéresserons uniquement aux méthodes de ML basées sur les nouvelles approches de Data Mining.

L'étude de ces articles a permis d'identifier 23 méthodes de ML utilisées pour effectuer une analyse prédictive. Ces 23 méthodes sont présentées dans le Tableau 4.4 avec le domaine d'application de chaque article. Il a été constaté que certains travaux utilisent une seule méthode pour effectuer la prédiction alors que d'autres associent plusieurs méthodes. Par ailleurs, certaines méthodes sont modifiées avant d'être utilisées alors que d'autres sont implémentées sans modification.

Tableau 4.4 : Méthodes de ML par article et domaine d'application

Article	**Méthode (s)**	**Domaine**
(Mojtaba, et al., 2018)	ANN, SVM, DT, AC	
(Tova M., et al., 2020)	Cox modele, DT	
(Lin, et al., 2019)	Analyse statistique (Hazard ratio (HR)	
(Beaulah Mary, et al., 2019)	MLR, MLR-WLS, Enhenced MLR	
(Gu-Wei, et al., 2019)	Cox Modele, MRMR, RF	
(Lee, et al., 2018)	ANN, SVM, RF, LDA, LR, GBM	
(Li, et al., 2019)	ANN, K-Means	
(Brian L., et al., 2019)	RF	

(M.Zolbanin & Dursu, 2018)	RF	Médecine
(Kaur & Kumari, 2018)	ANN, SVM, RBF, K-NN, DR	
(Bumberger, et al., 2019)	DT	
(Araz, et al., 2019)	ANN, SVM, RF, XGBoost, LR	
(Talaei-Khoei, et al., 2019)	ANN, SVM, DT, LR	
(Stephen, et al., 2019)	LR	
(Armando J, et al., 2018)	MAMLS,	
(Dharavath & Yogendra, 2019)	SVM, DT, RF, PCA, K-NN, NB	
(Hang, et al., 2019)	DT, RF, LR, GBM	
(Camila, et al., 2019)	DT, PCA, LDA, CA	Botanique
(Saurav, et al., 2019)	CA,	Science Sociales
(Dae-Hee, et al., 2020)	ANN, RF, GBM	
(José Rômulo, et al., 2019)	ANN, SVM, DT	
(Fenfen, et al., 2020)	XGBoost	
(José Rômulo, et al., 2019)	AMOS 24	
(Lima & Delen, 2019)	ANN, SVM, RF	
(M.Zolbanin & Dursu, 2018)	RF, LR, CT, ANOVA, PR, OLSR	
(Paul, et al., 2020)	BM	
(Ann & Yahia Mohamed, 2020)	ANN, Fuzzy Modele	Education
(Gray & Perkins, 2019)	DT	
(Fernandes, et al., 2019)	GBM	
(Jongko, et al., 2020)	DT, LR, RF, Adaboost	Construction

4.5.1. Méthodes de ML par technique d'apprentissage

Nous avons ensuite organisé les 23 méthodes identifiées par technique ou type d'apprentissage, sous-type d'apprentissage et domaine d'application. Le Tableau 4.5 donne le résumé de cette organisation. La section **Autres références,** contient des références potentielles pour plus de renseignement, sur les méthodes.

Tableau 4.5 : Méthodes de ML par technique d'apprentissage, articles et domaines d'utilisation

Méthode	Description	Type d'apprentissage	Sous Type	Articles	Domaine (s)	Autres références
ANN	Artificial Neural Network	Supervised and Unsupervised	Classification, Regression and clustering	(Abbas, et al., 2020), (Araz, et al., 2019), (Lima & Delen, 2019), (Kaur & Kumari, 2018), (Ann & Yahia Mohamed, 2020), (Lee, et al., 2018), (Li, et al., 2019), (Dae-Hee, et al., 2020), (Talaei-Khoei, et al., 2019) , (Dharavath & Yogendra, 2019)	Médecine, Sciences Sociales, Education	
SVM	Support Vector Machine	Supervised	Classification	(Lima & Delen, 2019) (Kaur & Kumari, 2018), (Abbas, et al., 2020), (Lee, et al., 2018), (Araz, et al., 2019), (Dae-Hee, et al., 2020), (Talaei-Khoei, et al., 2019), (Dharavath & Yogendra, 2019)	Médecine Sciences Sociales	

DT	Decision Tree	Supervised	Classification and Regression	(Lima & Delen, 2019), (Kaur & Kumari, 2018), (Abbas, et al., 2020), (Lee, et al., 2018), (Araz, et al., 2019), (Dae-Hee, et al., 2020), (Talaei-Khoei, et al., 2019), (Dharavath & Yogendra, 2019)	Médecine Construction Education Botanique	(Madhu, 2017)
AC	Auto Classifier Node	Supervised	Classification	(Abbas, et al., 2020)	Médecine	
MLR	Multi Linear Regression	Supervised	Regression	(Beaulah Mary, et al., 2019)	Médecine	
Cox	Cox modele	Supervised	Regression	(Tova M., et al., 2020), (Gu-Wei, et al., 2019)	Médecine	
MRMR	Maximum relevance minimum redundancy	Semi-Supervised	Dimentionnality Reduction	(Gu-Wei, et al., 2019)	Médecine	(Yan, et al., 2020)
RF	Random Forest	Supervised	Classification and Regression	(Lima & Delen, 2019), (Jongko, et al., 2020), (Tova M., et al., 2020), (Gu-Wei, et al., 2019), (Hang, et al., 2019), (Araz, et al., 2019), (Brian L., et al., 2019), (Shahid Mohammad & Majid, 2021), (Dharavath & Yogendra, 2019), (M.Zolbanin & Dursu, 2018)	Médecine Sciences Sociales Construction	
PCA	Principal Component Analysis	Unsupervised	Dimentionnality Reduction	(Camila, et al., 2019), (Dharavath & Yogendra, 2019),	Médecine Botanique	(Markus, 2008) (Chris, et al., 2006)

LDA	Linear Discriminant Analysis	Supervised	Classification	(Lee, et al., 2018), (Camila, et al., 2019)	Médecine Botanique	(Balakrishnama & Ganapathiraju, 1998)
CA	Cluster Analysis	Unsupervised	Clustering	(Camila, et al., 2019), (S. M. S. Basnet and al.2019)	Sciences sociales Botanique	
XGBoost	eXtreme Gradient Boosting algorithm	Supervised	Classification and Regression	(Araz, et al., 2019), (F. Ge and al.(2020))	Médecine, Sciences sociales	(Tianqi & Tong, 2020)
LR	Logistic Regression	Supervised	Classification	(Stephen, et al., 2019), (Jongko, et al., 2020), (Hang, et al., 2019), (Lee, et al., 2018), (Araz, et al., 2019) , (Talaei-Khoei, et al., 2019), (M.Zolbanin & Dursu, 2018)	Médecine Construction	(Zhu, 2016), (KALE, 2020)
GBM	Gradient Boosting Machine	Supervised	Regression and classification	(Hang, et al., 2019), (Lee, et al., 2018), (Shahid Mohammad & Majid, 2021), (Fernandes, et al., 2019)	Médecine	
FL	Fuzzy Logic	Supervised and Unsupervised	Classification, Regression, and clustering	(Ann & Yahia Mohamed, 2020)	Education	
RBF	Radial Basis Function	Supervised and Unsupervised	Classification, Regression, and clustering	(Kaur & Kumari, 2018)	Médecine	
K-NN	K-Nearest Neighbour	Supervised	Classification	(Kaur & Kumari, 2018), (Dharavath & Yogendra, 2019)	Médecine	(Pádraig & Sarah, 2021)
Adaboost	Adaptive boosting	Supervised	Classification	(Jongko, et al., 2020)	Construction	(MICHAEL, et al., 2002) (Solomatine & Durga, 2004)

K-Means	K-Means clustering	Unsupervised	Clustering	(Li, et al., 2019)	Médecine	
NB	Naïve Bayes	Supervised	Classification	(Dharavath & Yogendra, 2019)	Médecine	(Zhang, 2004)
PR	Poisson Regression	Supervised	Regression	(M.Zolbanin & Dursu, 2018)	Sciences Sociales	
OLSR	Ordinary Least Squares Regression	Supervised	Regression	(M.Zolbanin & Dursu, 2018)	Sciences Sociales	
BM	Bayesian methods	Supervised and Unsupervised	Regression and Clustering	(Paul, et al., 2020)	Sciences Sociales	

Tel que dit ci-haut, nous distinguons quatre techniques d'apprentissage. Il a été observé au cours de cette étude, que l'apprentissage supervisé est la technique la plus utilisée quand il s'agit de faire une analyse prédictive. Nous pouvons le constater sur la Figure 4.4 qui donne les taux d'utilisation des différentes techniques d'apprentissage pour les articles étudiés. 70% des méthodes utilisées pour effectuer l'analyse prédictive sont de type supervisé, ensuite vient l'apprentissage non supervisé et très faiblement l'apprentissage semi-supervisé. Aucun auteur n'a utilisé l'apprentissage par renforcement pour faire de la prédiction.

Cette observation permet d'avoir un premier indice sur le choix du type d'apprentissage pour faire une analyse prédictive.

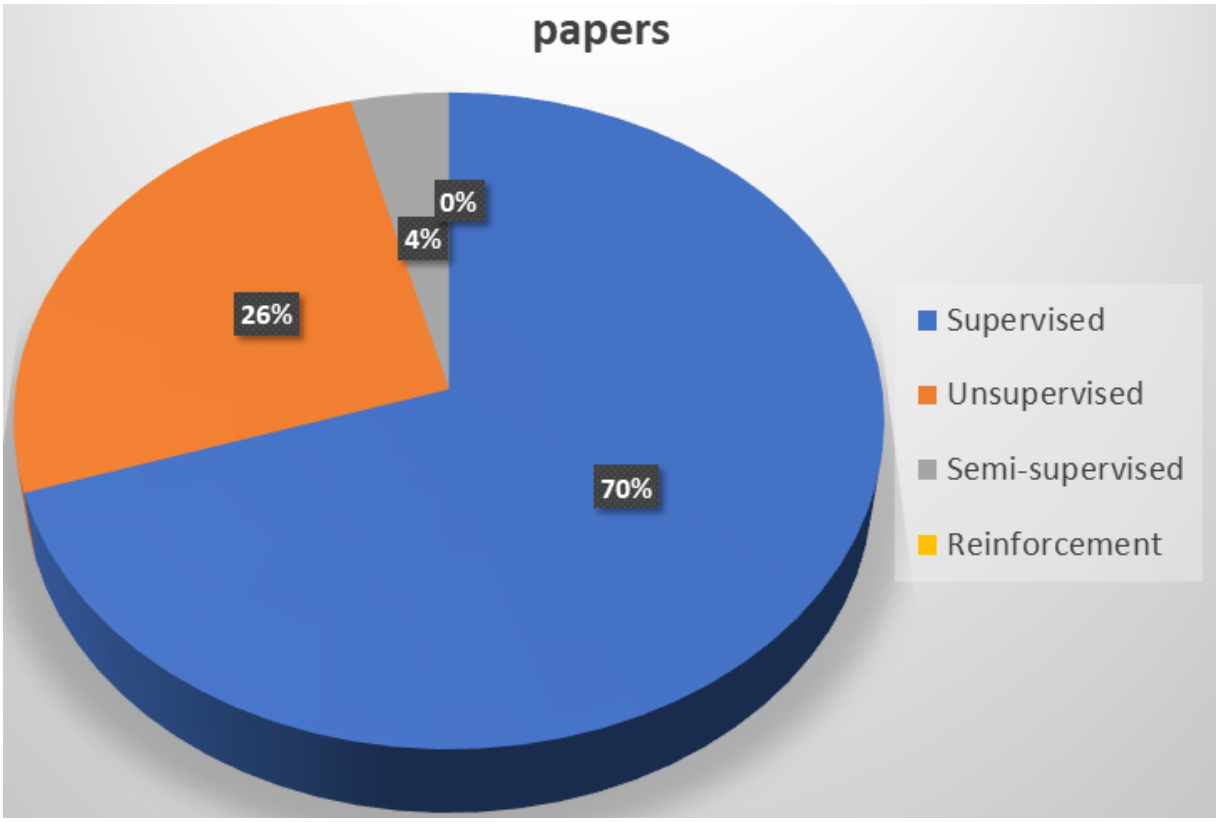


Figure 4.4 : Techniques ML utilisées pour l'analyse prédictive dans les papiers étudiés

4.5.2. Domaine d'application par méthodes de ML

Un autre indice, dans le choix de la méthode de Machine Learning (ML) à utiliser peut-être trouvé au niveau des domaines d'application desdites méthodes. La Figure 4.5 permet de déterminer les domaines d'application des méthodes de ML pour une analyse prédictive. Nous y observons que la médecine est la discipline la plus utilisée comme domaine d'application des

méthodes de ML pour l'analyse prédictive. Mais il n'y a pas que la médecine, il y a aussi les sciences sociales, le suivi des constructions, ensuite la botanique et enfin l'éducation.

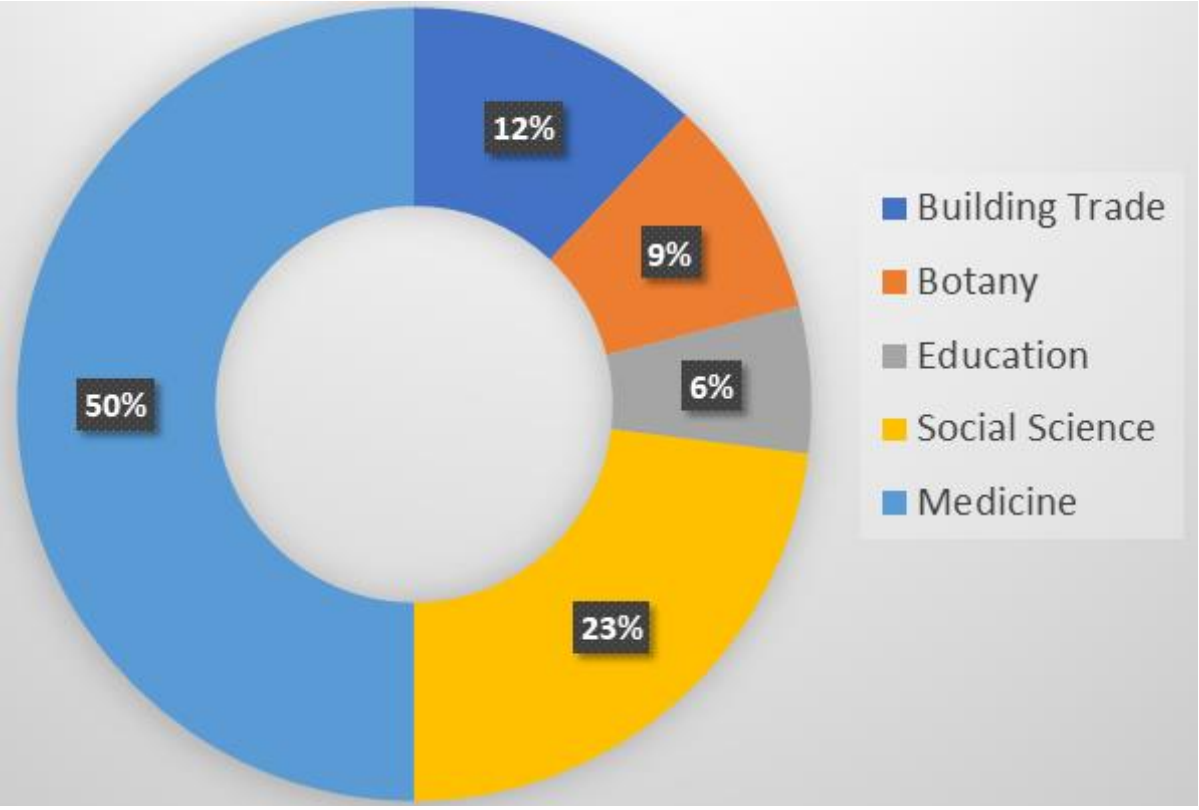


Figure 4.5 : Domaines d'application des méthodes de ML pour une analyse prédictive

Il faudra noter que l'utilisation d'une méthode dans un domaine n'est pas exclusive. Une méthode peut être utilisée dans plusieurs domaines. Nous voyons par exemple l'ANN qui est utilisée en médecine (Abbas, et al., 2020), (Araz, et al., 2019), (Lima & Delen, 2019) (Kaur & Kumari, 2018), (Ann & Yahia Mohamed, 2020), (Lee, et al., 2018), (Li, et al., 2019), (Dae-Hee, et al., 2020), (Talaei-Khoei, et al., 2019), (Dharavath & Yogendra, 2019), en sciences sociales (Dae-Hee, et al., 2020), (José Rômulo, et al., 2019), (Paul, et al., 2020) et dans l'éducation (Ann & Yahia Mohamed, 2020) ; ou alors le CA qui est utilisé à la fois en botanique (Kaur & Kumari, 2018) et en sciences sociales (Saurav, et al., 2019).

Il nous paraît donc important d'avoir une vue générale sur les méthodes de machine Learning les plus utilisées pour faire de l'analyse prédictive, tous domaines confondus, d'où la Figure 4.6.

Grâce à la Figure 4.6, il apparait clairement que le RF est la méthode de ML la plus utilisée pour les analyses prédictives, tous domaines confondus, suivi d'ANN et DT, ensuite SVM puis

LR et les autres. Ceci dit, bien que RF soit la méthode la plus utilisée, il est intéressant de remarquer son absence dans certains domaines, tels que l'éducation et la botanique. Il en est de même pour ANN qui n'est pas utilisé en construction et en botanique, DT qui n'est pas utilisé en sciences sociales et en botanique, SVM qui n'est pas utilisé en construction, botanique et éducation, ainsi que LR qui n'est pas utilisé dans l'éducation et la botanique.

Ainsi, pour effectuer un bon choix, il faut considérer l'utilisation des méthodes dans chaque domaine. Voilà pourquoi la Figure 4.7 donne le taux d'utilisation des cinq principales méthodes ML les plus utilisées pour l'analyse prédictive par domaine d'application. Ces cinq méthodes couvrent à elles seules 22 des 30 articles, ainsi que les cinq domaines d'application répertoriés dans cette étude.

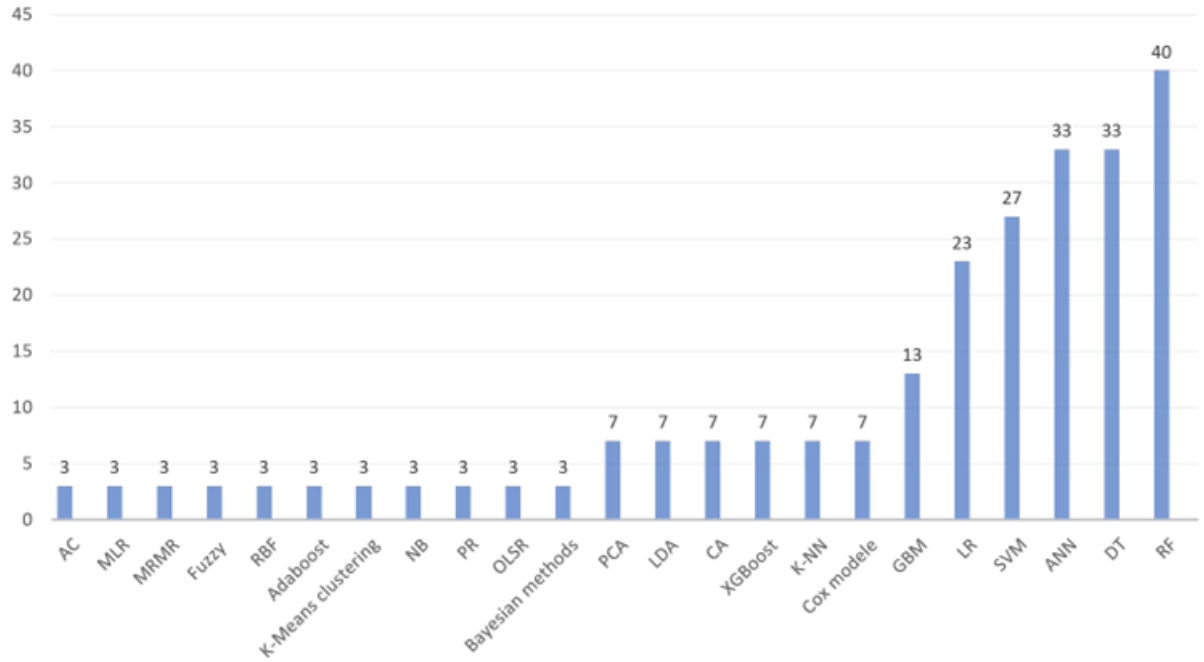


Figure 4.6 : Taux d'utilisation des méthodes ML dans les papiers étudiés.

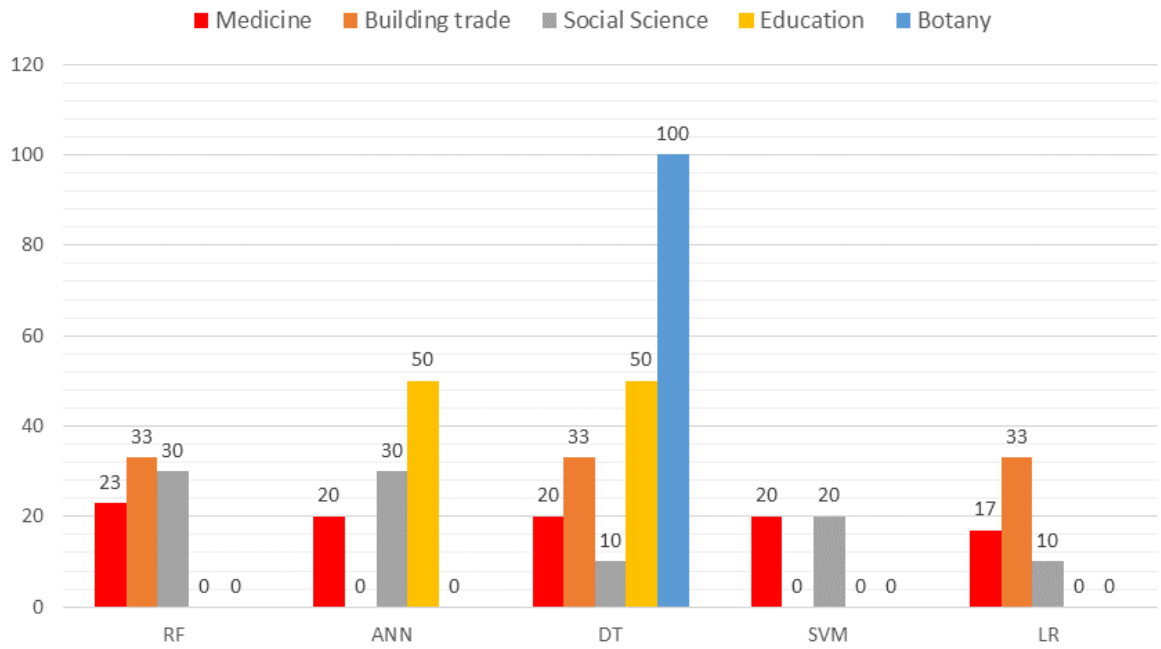


Figure 4.7 : Taux d'utilisation des 5 principales méthodes ML par domaine

La Figure 4.7 permet de constater, pour les articles étudiés dans cette revue, que :

- La méthode RF est la plus utilisée en médecine ;
- ANN et DT sont les plus utilisées dans l'éducation ;
- DT est la seule méthode utilisée dans le seul article de botanique étudié dans cette revue ;
- RF a le même taux d'utilisation qu'ANN dans les sciences sociales ;
- LR, RF et DT ont le même taux d'utilisation dans la construction.

4.5.3. Classification VS Régression

L'objectif de cette analyse serait de prédire les contaminées de l'épidémie, réduire les faux positifs et contribuer ainsi à limiter la propagation de l'épidémie, tout en tenant compte des autres paramètres tels que la région, la période ainsi que la marge d'erreur.

Ainsi, en tenant compte des deux éléments suivants :

1. Le résultat de la sous-section 4.5.2 qui montrent que l'apprentissage supervisé est le type d'apprentissage le plus utilisé pour effectuer une analyse prédictive ;
2. Le fait que « *L'apprentissage non supervisé se caractérise par l'absence d'une connaissance exacte de ce que contiennent les données ou* ***de l'objectif final*** » (Moujahid, et al., 2018). Chose contraire à notre cas, où l'objectif final est bien connu ;

la suite de cette revue de littérature se focalisera sur les méthodes de l'apprentissage supervisé relevées dans cette étude, notamment : *ANN, SVM, DT, AC, MLR, Cox Modele, RF, LDA, XGBoost, LR, GBM, Fuzzy, RBF, K-NN, Adaboost, NB, PR, OLSR et Bayesian methods.* Le Tableau 4.6 ne retient du Tableau 4.5 que les méthodes de l'apprentissage supervisé. Il donne ainsi la liste de toutes les méthodes de l'apprentissage supervisé utilisées au sein des articles étudiés regroupées en deux sous-types d'apprentissage (classification et régression).

Pour distinguer ces deux sous-types d'apprentissage, rappelons la différence entre classification et régression que propose (Helin, et al., 2020) : La classification consiste à trouver une relation entre les entrées et ***les sorties discrètes***. Les variables de sorties sont également appelées ***catégories ou étiquettes*** (Helin, et al., 2020)**.** La régression quant à elle, consiste à estimer ou à prédire ***des quantités continues*** (Helin, et al., 2020).

Tableau 4.6 : Méthodes de l'apprentissage supervisé par sous-type d'apprentissage

Sous-type d'apprentissage	Méthodes
Classification	SVM, AC, LDA, LR, Fuzzy, RBF, K-NN, Adaboost et NB
Régression	PR, OLSR, Bayesian Methods, MLR et Cox Modele
Classification & Régression	ANN, DT, RF, XGBoost et GBM

En effet, les deux groupes (classification et régression) permettent de construire un modèle de prédiction d'une valeur à partir des données entrées en se servant des modèles d'entrées/sorties préalablement saisis. Mais il y a cependant deux points de différence entre les deux :

- La valeur prédite est numérique pour la régression et catégorielle pour la classification (Zhu & Fang, 2016) (Sparrow, 2022).
- La régression permet de prédire une valeur continue, c'est-à-dire qui peut prendre en théorie une infinité de valeurs formant un ensemble continu (Bradley & & Schapire, 2007), alors que la classification permet de prédire une valeur discrète, c'est-à-dire qui a une valeur finie, qu'il est possible d'énumérer (1, 2, 3, etc.) (Bradley & & Schapire, 2007). En d'autres termes, une valeur discrète appartient à un ensemble déterminé d'éléments alors qu'une valeur continue appartient à un intervalle.

4.5.3.1. Classification

Un problème est dit de classification lorsque la sortie (le résultat) est une catégorie telle que 'rouge' ou 'bleu' ou alors 'maladie' ou 'pas de maladie'. Un modèle de classification tente de tirer des conclusions à partir des valeurs observées (Zhu, 2016).

Voici quelques caractéristiques des problèmes de classification (Sparrow, 2022) :

- Un problème de classification exige que les exemples soient classés dans l'une des deux ou plusieurs classes ;
- Un problème à deux classes est souvent appelé en anglais un 'two class classification problem' ou un 'binary classification problem' ;
- Un problème à plus de deux classes est souvent appelé en anglais un 'multi class classification problem' ;
- Un problème où un exemple appartient à plusieurs classes (catégorie) est appelé un 'multi-label classification problem'.

Exemples de problèmes de classification :

- Prédire l'origine botanique et géographique du miel, (Camila, et al., 2019) ;
- Prédire les résultats thérapeutiques de la dépression, (Lee, et al., 2018) ;
- Prédire les familles à faible revenu, (José Rômulo, et al., 2019).

Il arrive assez souvent que le résultat d'une classification soit sous forme de probabilités d'appartenir à plusieurs groupes. Dans ce cas, l'exemple doit être considéré comme appartenant au groupe ayant la plus grande probabilité.

L'une des manières d'estimer la qualité d'un modèle prédictif de classification est de calculer sa précision (accuracy). La précision de classification est le pourcentage d'exemples correctement classés parmi toutes les prédictions faites.

$$\text{Précision} = \frac{\text{Prédiction correctes}}{\text{Total des prédictions}} * 100$$

4.5.3.2. Régression

Un problème est dit de régression lorsque le résultat attendu est une valeur réelle ou continue, comme *le salaire* ou *le poids* (Zhu & Fang, 2016). En d'autres termes les possibilités sont quasi-illimitées.

Voici quelques caractéristiques des problèmes de régression (Sparrow, 2022) :

- Un problème de régression nécessite la prédiction d'une quantité ;
- Un problème avec des variables d'entrée multiple est souvent appelé un problème de régression multivariée ;
- Un problème de régression où les variables d'entrée sont ordonnées par le temps est appelé un problème de prévision de séries chronologiques.

Exemples de problèmes de régression :

- La prédiction de l'efficacité des médicaments (traitements) contre le cancer buccal (OSCC) (Lima & Delen, 2019) ;
- La prédiction de la récidive dans les tribunaux de la toxicomanie (M.Zolbanin & Dursu, 2018) ;
- La prédiction de la récidive d'un carcinome hépatocellulaire après résection (Gu-Wei, et al., 2019).

La qualité d'un modèle prédictif de régression ne se mesure pas par le calcul de la précision comme dans le cas de la classification, mais par celui de l'erreur quadratique moyenne, RMSE en anglais *Root Mean Square Error.*

4.6. Choix de la méthode ML pour la prédiction des épidémies

Dans une suite logique des travaux de recherche liés à la thématique de cette thèse, dans le but de choisir la (les) méthode (s) d'analyse prédictive les plus appropriées dans un processus d'enquête épidémiologique, nous avons procédé à une revue de littérature.

Cette revue de littérature présentée dans les sections précédentes, donne une évaluation des tendances des méthodes de Machine Learning (ML) utilisées dans l'analyse prédictive. Pour fournir une revue qui reflète la situation actuelle de la recherche, nous avons adopté une démarche qui a permis de retenir uniquement les travaux de recherche les plus récents dans la littérature.

Cela a permis d'identifier :

- Les disciplines les plus utilisées comme domaine d'application des méthodes de ML pour l'analyse prédictive : la médecine, les sciences sociales, la construction, la botanique et l'éducation ;
- Les techniques d'apprentissage les plus utilisées : Supervisé et Non supervisé ;
- Les méthodes de ML les plus utilisées tous domaines confondus : RF, ANN, DT, SVM et LR ;
- Et les méthodes de ML les plus utilisées par discipline : DT et ANN dans l'éducation, LR, RF et DT dans la construction, DT dans la botanique, RF et ANN dans les sciences sociales et RF dans la médecine.

Ce résultat ainsi présenté permet, non pas de dicter quelle méthode choisir mais, de savoir quelles sont les méthodes que les autres chercheurs dans le monde utilisent dans une discipline précise pour faire de l'analyse prédictive. Ce qui peut servir d'indicateur pour les chercheurs qui s'intéressent à l'analyse prédictive.

Bien que notre principale contribution se positionne dans le sous type *Classification,* nous avons opté pour les méthodes pouvant faire à la fois la classification et la régression. Ceci pour ne pas restreindre notre champ d'action et conserver ainsi la possibilité de prédire des quantités continues (régression) si nécessaire.

A ce stade, le fait que la méthode RF apparait à la fois dans la liste des méthodes les plus utilisées (tous domaines confondus) et aussi comme la méthode la plus utilisée en médecine (selon les articlesétudiés) est un indicateur intéressant pour le choix de la méthode à utiliser dans le cadre de cette thèse. En plus de cela, RF est une méthode très souvent utilisée pour la classification mais peut aussi être utilisée pour la régression, ce qui répond à notre exigence de ne pas fermer la porte à d'éventuelles prédictions de régression.

Ceci dit, nous ne nous sommes pas contentés de cet état de l'art pour sélectionner RF comme la méthode à utiliser dans cette thèse. Dans cette section, pour confirmer ou réfuter cette idée nous avons mené une brève revue de littérature orientée vers la prédiction dans le domaine **épidémiologique**. Cette brève revue de littérature vient approfondir celle effectuée dans les cinq sections précédentes et elle a lieu parce que cette thèse de doctorat s'intéresse aux enquêtes mobiles **épidémiologiques.**

Dans les sections précédentes, comme le montre la Figure 4.7, nous avons conclu que les méthodes les plus utilisées en médecine sont : RF, ANN, DT, SVM et LR.

Dans la littérature nous retrouvons des travaux de plusieurs chercheurs sur la prédiction des épidémies. Pour cette étude, nous avons analysé les travaux de recherche portant sur la prédiction de quatre épidémies notamment :

- La peste porcine Africaine ;
- La dengue ;
- La grippe ;
- Le Norovirus de l'huître.

4.6.1. Les épidémies considérées

4.6.1. 1. La Peste Porcine Africaine

La Peste Porcine Africaine (PPA) est une infection grave qui, à ce jour, n'a aucun vaccin. Elle concerne les porcs domestiques, les sangliers et les suidés sauvages africains (phacochères et potamochères) (Remongin, 2022). La PPA est présente en Afrique subsaharienne, dans les Pays Baltes (Estonie, Lituanie, Lettonie), en Pologne, Ukraine, Russie, Moldavie, Roumanie, Italie (Sardaigne), Bulgarie, Belgique, Serbie, Hongrie, Slovaquie et Allemagne, mais aussi dans certains pays d'Asie : Birmanie, Chine, Inde, Corée du Nord et du Sud, Cambodge, Vietnam, Philippines, Indonésie, Laos, Mongolie et Timor-Est (Remongin, 2022).

Elle cause beaucoup de tort tant aux animaux qu'aux industriels. Après une infection par des souches virulentes, le taux de mortalité des porcs malades est proche de 100 % (Liang, et al., 2020). Face à une telle épidémie, une analyse prédictive peut être salvatrice. Elle permettrait aux éleveurs et industriels de prendre les mesures nécessaires pour limiter les pertes et la souffrance des animaux.

Les méthodes de ML utilisées en association avec les données météorologiques afin de fournir une prévision de la peste porcine au sein de l'étude analysée (Liang, et al., 2020) sont : Naive Bayes (NB), Random Forest (RF), Adaboost, Support Vector Machine (SVM), Artificial Neural Network (ANN) et C4.5 sur les prédicteurs suivants : le moment de l'apparition de la maladie, la longitude et la latitude du lieu de l'épidémie ainsi que le nombre d'animaux touchés hormis les données météorologiques.

4.6.1.2. La dengue

La dengue est l'une des maladies virales transmises par des vecteurs les plus célèbres, car elle est propagée par des moustiques infectés. Elle est transmise particulièrement par les moustiques femelles (Aedes Aegypti) dans les régions tropicales et subtropicales du monde entier (Naiyar & Mohammad, 2019), (Anno, et al., 2019).

Selon des estimations issues d'une modélisation, 390 millions infections par le virus de la dengue surviennent chaque année, dont 96 millions se manifestent cliniquement, tous degrés de gravité confondus, (Dengue, 2022). Le risque d'infection existe dans 129 pays (Brady, 2012) mais 70 % de la charge réelle pèse sur l'Asie (Dengue, 2022). Le nombre de cas de dengue notifiés par l'OMS a été multiplié par plus de huit au cours des deux dernières décennies, passant de 505 430 cas en 2000 à plus de 2,4 millions de cas en 2010 et 4,2 millions de cas en 2019. Le nombre de décès déclarés entre 2000 et 2015 est passé de 960 à 4032 (Dengue, 2022).

Fort de ce constat, plusieurs travaux de recherche ont été effectués afin de contribuer à lutter contre cette épidémie (Anno, et al., 2019) (Dhesi Baha, et al., 2019), (Naiyar & Mohammad, 2019).

Les méthodes ML utilisées pour prédire la dengue au sein des articles étudiés sont les suivantes : Bayesian Network (BN), Convolutional Neural Network (CNN), Logistic Regression (LR), k-Nearest Neighbour (kNN), SVM, RF, ANN, C4.5, CART et NB sur les prédicteurs suivants : la température, la température de surface de la mer, l'humidité, les précipitations, la densité de la population, la date d'apparition et de notification ainsi que les indices vectoriels tels que le nombre d'Aedes. albopictus, le nombre d'Aedes. aegypti et le nombre de larves.

4.6.1.3. La grippe

La grippe est une infection respiratoire aiguë, due à un virus Influenza. Les maladies de type grippal, ILI (Infections Respiratoires Aiguës) sont considérées comme les causes les plus importantes de mortalité dans le monde (Tapak, et al., 2019). Saisonnières, les épidémies de grippe touchent tous les pays du globe. Les pays à quatre saisons (le Maroc, la France, etc.) sont souvent affectés par ces épidémies pendant l'hiver (Lexpress, 2018)et ceux à deux saisons (la RDC, le Rwanda, le Sénégal, le Ghana etc.) pendant la saison sèche bien que présente aussi pendant la saison de pluie (OMS, 2020).

Il existe 3 types de grippes saisonnières – A, B et C. Les virus grippaux de type A se subdivisent en sous-types. Parmi les nombreux sous-types des virus grippaux A, les sous-types A(H1N1) et A(H3N2) circulent actuellement chez l'homme (OMS, 2018). En France, l'épidémie de grippe saisonnière dont le H1N1 survient chaque année, généralement entre les mois de novembre et avril et dure en moyenne 10 à 11 semaines (Santé Publique, 2019).

Les chercheurs informaticiens spécialistes en ML ne sont pas restés indifférents face aux épidémies grippales. Plusieurs études (Tapak, et al., 2019), (Koike & Morimoto, 2018) ont été menés ayant pour but de prédire une épidémie grippale utilisant les méthodes ML suivantes : ANN, RF et SVM sur les prédicteurs suivants : la période, l'année, la saison, l'humidité, la vitesse du vent et la température.

4.6.1.4. Le norovirus de l'huitre

Le norovirus est la principale cause de maladie et d'épidémies dues à des aliments contaminés tels que les huîtres récoltées dans les eaux côtières contaminées par les eaux usées (Shima Shamkhali & Zhiqiang, 2017). Même si le norovirus peut se transmettre d'une personne à une autre, le moyen de transmission le plus fréquent est l'alimentation avec en tête de fil les huitres (Shima & Deng, 2018).

Le norovirus est associé à 18 % de toutes les maladies diarrhéiques dans le monde et est à l'origine de 58 % des maladies d'origine alimentaire, soit 5,5 millions de cas au cours d'une année aux États-Unis (Shima Shamkhali & Zhiqiang, 2017). Plusieurs pays ont été affectés par cette épidémie dont le Pays-Bas, le Japon, l'Italie, etc. (Adeline, et al., 2011).

Les études de prédiction de cette épidémie ont utilisé les méthodes ML suivantes : ANN, RF, LR et Genetic Programming (GP) sur les prédicteurs suivants : période de l'épidémie, nombre des huitres contaminées, la température de l'eau, le rayonnement solaire, la hauteur des jauges, la salinité, le vent et les précipitations.

4.6.2. Les méthodes les plus utilisées et leurs précisions par épidémie

Une observation des méthodes utilisées pour la prédiction des épidémies mentionnées précédemment, fait ressortir une liste de six méthodes comme étant les plus utilisées. Il s'agit de LR, C4.5, NB, SVM, ANN et RF. Ces six méthodes ont été utilisées par des chercheurs indépendants des pays différents pour prédire au moins deux épidémies.

De cette liste de six méthodes les plus utilisées dans la prédiction des épidémies, nous constatons que trois d'entre elles sont plus utilisées que les autres. Il s'agit de RF, ANN et SVM. Ces trois méthodes ont été utilisées pour prédire au moins trois épidémies différentes. Enfin, deux méthodes sortent du lot : RF et ANN. Elles ont été utilisées pour toutes épidémies considérées dans cette étude et dans la grande majorité des travaux étudiés.

Le Tableau 4.7 donne une comparaison des précisions des méthodes ML dans la prédiction des épidémies considérées dans cette étude. Nous y remarquons par exemple que pour la PPA la méthode ayant la plus grande précision est RF suivi de C4.5; pour la dengue c'est RF et LogitBoost avec la même précision ; pour la grippe c'est SVM suivi de l'ANN ; pour le norovirus de l'huitre c'est ANN suivi de RF, puis LR et GP qui ont la même précision.

La méthode RF sort du lot en se positionnant comme la méthode ayant la meilleure précision pour deux épidémies sur quatre.

Tableau 4.7 : Comparaison des résultats de précision des méthodes ML dans la prédiction des épidémies

	La PPA	**La dengue**	**La grippe**	**Le norovirus**
ANN	97,04% (Liang, et al., 2020)	81% (Anno, et al., 2019)	88, 9% (Tapak, et al., 2019)	99, 83% (Shima & Deng, 2018)
LogitBoost	-	92% (Naiyar & Mohammad, 2019)	-	-
RF	98,43% (Liang, et al., 2020)	92% (Fathima & Manimeglai, 2015)	86, 5% (Tapak, et al., 2019)	98, 82 % (Shima Shamkhali & Zhiqiang, 2017)
SVM	91,84% (Liang, et al., 2020)	90,42% (Fathima & Manimegalai, 2012)	89,2% (Tapak, et al., 2019)	-

C4.5	97,36% (Liang, et al., 2020)	84,7% (Tanner, et al., 2008)	-	-
LR	-	62% (Agarwal, et al., 2018)	-	88, 82% (Shima Shamkhali & Zhiqiang, 2017)
NB	95,50% (Liang, et al., 2020)	84% (Dhesi Baha, et al., 2019)	-	-
kNN	-	-	-	-
Adaboost	95,20% (Liang, et al., 2020)	-	-	-
GP	-	-	-	88,82% (Shima Shamkhali & Zhiqiang, 2017)

4.7. Conclusion

L'analyse faite par Yuexin Wu et ses co-auteurs dans (Wu, et al., 2018) *: « La nature temporelle des données épidémiologiques ainsi que la nécessité d'une prédiction en temps réel font que le problème de prédiction épidémiologique se classe dans la catégorie de prédiction de séries chronologiques* (Wu, et al., 2018)*».*

Maintenant que nous avons classé la prédiction des épidémies dans la catégorie de prédiction de séries chronologiques. Une analyse poussée des travaux de Leili Tapak et all (Tapak, et al., 2019), de Ruirui Liang et all (Liang, et al., 2020) ainsi que de Gilles R (Ducharme, 2018) nous permet d'identifier la méthode ML la plus adaptée pour les problèmes de type séries chronologiques.

En effet, Leili Tapak et ses co-auteurs font une comparaison de trois méthodes ML (SVM, ANN et RF) afin de juger leur capacité de prédiction pour les problèmes de type séries chronologiques (les épidémies). Ils en viennent à la conclusion selon laquelle la prédiction temporelle du RF est meilleure par rapport à celles des autres méthodes étudiées (ANN et SVM) pour des problèmes de ce type alors que ANN est meilleure dans la détection des épidémies. Liang et ses co-auteurs testent et comparent six méthodes ML (Bayes Net, ANN, SVM, AdaBoost, C4.5 et RF) sur la base de quatre mesures de comparaison bien connues : SN (sensitivity), SP (specificity), ACC (accuracy) & MCC (Matthew Correlation Coefficient) avant d'opter pour le RF pour effectuer l'analyse prédictive de l'épidémie de la Peste Porcine en Afrique, parce que le résultat de cette comparaison a placé le RF au-dessus des autres. Gilles R. Ducharme compare six méthodes (Bn, RLog, SVM, RF, KNN, ANN). A l'issue de son étude, Gilles R. Ducharme parvient à la conclusion suivante : « Le classifieur qui ressort de cet exercice avec les meilleurs scores est RF et ses variantes. ».

Dans la sous-section précédente, nous avions retenues ANN et RF comme étant les deux méthodes les plus utilisées pour la prédiction des épidémies. Dans celle-ci, nous pouvons remarquer que ANN et RF font à chaque fois partie des méthodes comparées par plusieurs chercheurs indépendants pour la prédiction des épidémies, et c'est souvent la méthode RF qui donne de meilleurs résultats.

Fort de tout ceci, la méthode Random Forest (RF) est recommandée comme méthode à utiliser pour la prédiction des épidémies sans négliger les réseaux de neurones artificiels (ANN).

En plus, nous avions considéré quatre épidémies dans cette étude : la PPA, la dengue, la grippe et le norovirus de l'huitre. L'analyse des études considérées a montré que pour la PPA la méthode ayant la plus grande précision est RF avec une précision de 98,43% suivi de C4.5 avec une précision de 97,36%; pour la dengue c'est RF et LogitBoost avec la même précision 92% ; pour la grippe c'est SVM avec 89,2% suivi de l'ANN avec 88,9% ; pour le norovirus de l'huitre c'est ANN avec 99, 83% suivi de RF, LR et GP qui ont la même précision, soit 88,82%.

La méthode RF se démarque des autres à la fois par sa présence dans la prédiction de toutes les épidémies considérées dans cette étude mais aussi par sa précision assez élevée pour toutes ces épidémies. Nous l'avons ainsi choisi pour la suite de cette thèse.

Chapitre 5 : Formalisation des forêts aléatoires

5.1. Introduction

Dans le chapitre 4 consacré à la revue de littérature des méthodes de Machine Learning pour l'analyse prédictive, nous avons conclu que la méthode Random Forests, que nous traduisons en français par forêts aléatoires, est probablement la plus adaptée pour des problématiques de séries chronologiques (ou time series) dont fait partie l'analyse prédictive épidémiologique à laquelle s'intéressent les travaux de cette thèse. Afin de mieux comprendre le fonctionnement de cette méthode, cerner les problématiques liées à son utilisation et ainsi proposer des solutions empiriques, nous avons jugé nécessaire d'effectuer une étude de formalisation. C'est ce travail de formalisation qui est présenté dans ce chapitre.

Léo Breiman, initiateur à l'origine de Random Forests (RF) définit cette méthode de la manière suivante : « Les forêts aléatoires sont une combinaison de prédicteurs d'arbres telle que chaque arbre dépend des valeurs d'un vecteur aléatoire échantillonné indépendamment, et avec la même distribution pour tous les arbres de la forêt (Breiman, 2001)».

A ce stade un élément important à noter dans cette définition originelle du RF est qu'il s'agit d'un ensemble (combinaison) d'arbres. Ceci implique que sa compréhension passe inexorablement par celle du fonctionnement des arbres de décision. Et d'ailleurs toute son efficacité vient de celui des arbres de décision.

La section 2 de ce chapitre sera consacrée à la compréhension des arbres de décision allant de leur définition à leur fonctionnement en passant par leur jargon.

En outre, la méthode Random Forests fait partie des méthodes multi-classifieurs. Le paradigme de ce type de méthodes est de faire intervenir, à titre égal, plusieurs algorithmes pour prendre une décision. C'est comme si, pour un projet donné, nous faisons intervenir plusieurs experts indépendants qui donneront chacun son avis avant la prise de la décision finale.

Une taxonomie très répandue dans la littérature (Giacinto & Roli, 2001) classe ces méthodes multi-classifieurs en deux catégories : les multi-classifieurs homogènes (qui assemblent les classifieurs de même type, par exemple des multi-classifieurs constitués uniquement des arbres de décision, ou uniquement des réseaux de neurones) et les multi-classifieurs hétérogènes (qui assemblent des classifieurs de différents types). La méthode Random Forests fait ainsi partie de la première catégorie des multi-classifieurs souvent désigné par le sigle EoC pour *Ensemble of Classifiers*, en français Ensemble de Classifieurs, parce qu'elle combine uniquement les classifieurs de type arbre de décision.

La section 3 sera réservée aux Ensembles de Classifieurs. Nous y présenterons les raisons qui ont poussé les chercheurs à proposer les EoC malgré l'efficacité des arbres de décision avant de présenter les différents types et architectures de composition des EoC. Toujours dans cette section, nous verrons aussi quelques principes d'induction des EoC. Il s'agit là des principes sur la base desquels les EoC sont fondés. Leur connaissance est incontournable à la bonne compréhension de ces derniers, et donc à celle des RF.

Dans la section 4, nous donnerons les algorithmes d'induction des forêts aléatoires. Nous en avons retenu deux. Cependant, avant de donner les deux algorithmes d'induction, la première partie de cette section se proposera de clarifier les principes inhérents au fonctionnement de la méthode Random Forests introduite par Leo Breiman et aussi de parler des problématiques liées à leur mise en œuvre.

La section 5 qui conclut ce chapitre rappellera les grandes lignes de cette étude ainsi que la direction que prendra la solution que nous allons proposer dans le chapitre suivant.

5.2. Les arbres de décision

En Machine Learning la méthode arbre de décision (DT) porte ce nom par analogie avec l'arbre naturel (botanique) que nous pouvons avoir dans notre jardin. De même que son nom, la terminologie des arbres de décision s'inspire des différentes parties d'un arbre. Cette terminologie contient trois principaux termes (Crucianu, et al., 2021) :

- Nœud racine : Il désigne le nœud de départ où se trouve toutes les données à la construction de l'arbre, à l'image de la racine d'un arbre qui est le point de départ de la germination d'un grain.
- Nœuds internes : Désignent des nœuds qui viennent du nœud racine ou des autres nœuds. Ils sont appelés nœuds internes au moment où ils ne sont ni nœud racine ni des feuilles. Ils sont donc à l'intérieur de l'arbre.
- Les feuilles : Désignent des nœuds qui n'ont pas de descendant. Ce sont elles qui contiennent les décisions finales dites étiquètes dans le cas de la classification.

En plus de ces trois termes, il n'est pas rare d'utiliser le terme *branche* dans le jargon des arbres de décision. Une branche, comme sur arbre naturel, désigne une suite des nœuds et feuilles ayant le même nœud de départ, après le nœud racine.

Selon Quinlan (Quinlan, 1996) **,** un arbre de décision est soit un nœud étiqueté soit un nœud test lié à au moins deux branches. Afin de représenter hiérarchiquement les données d'un

ensemble (une base de données) et arriver à des prédictions, le nœud test (nœud racine) calcule des résultats basés sur un attribut. Chaque résultat correspond à une branche. Ainsi de suite jusqu'à l'obtention des feuilles qui porteront les prédictions finales.

Murthy, (Murthy, 1996) définit les arbres de décision comme un moyen de représenter un ensemble de règles qui présentent les données par une structure hiérarchique, en partitionnant de façon récursive l'ensemble de données. Sur la Figure 5.1 nous donnons un exemple d'un arbre de décision. Cet arbre de décision représente une partie du dataset que nous avons utilisé dans le cadre de cette thèse dans un article de recherche qui sera publié incessamment. Il s'agit d'un dataset qui traite de la situation de la covid-19 au Maroc. Le DT sur la Figure 5.1 représente les données suivantes :

- Sur la première ligne : toutes les données de la base de données dans un seul nœud.
- Sur la deuxième ligne : les données séparées selon les dates.
- Sur la troisième ligne : A la date du 22 mars 2020, nous avons d'une part le nombre total des cas covid-19, à gauche de l'écran (main gauche de l'utilisateur en position de saisie), d'autre part le nombre des tests effectués, à droite de l'écran. Idem pour le 22 avril 2020.
- Quatrième ligne : de la gauche vers la droite le nombres des nouveaux cas covid-19, de morts et de personnes rétablies, comme nœuds fils du nœud « Total Cas », puis le nombre de nouveaux tests comme nœud fils du nœud « Total Tests ».

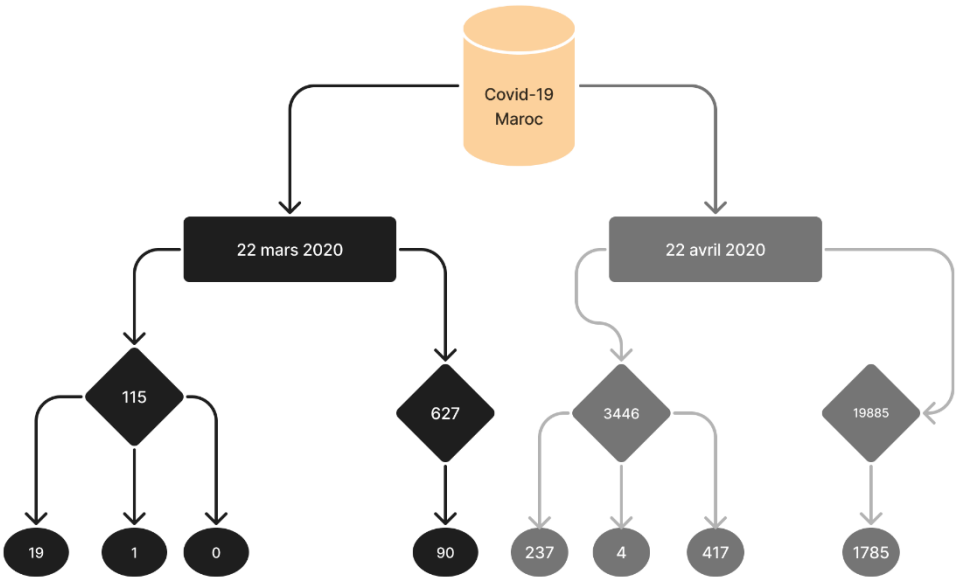


Figure 5.1 : Représentation sur un arbre de décision du dataset sur la situation de la covid-19 au Maroc

Il existe plusieurs algorithmes d'induction d'arbre de décision, les plus connus sont ID3, C4.5 et CART. Ces principaux algorithmes utilisent deux phases pour construire un arbre de décision

optimal : la phase d'expansion d'une part ainsi que la phase d'élagage et attribution des conclusions d'autre part (Shamrat, et al., 2022).

1. La phase d'expansion

Aussi appelée phase d'apprentissage ou phase de construction, elle consiste à construire la structure hiérarchique de l'arbre suivant un processus récursif de division de l'espace de données en sous-régions (nœuds) de plus en plus homogène. Le principe de cette phase est de placer, pour commencer, tous les éléments (aussi appelés observations) dans le même nœud, le nœud racine. Par la suite le nœud racine sera découpé en plusieurs nœuds sur la base de variables (caractéristiques) qu'il contient. Les nouveaux nœuds ainsi créés sont appelés nœuds fils par rapport au nœud racine qui est le nœud père. Les données sont par la suite reparties dans les nœuds fils selon qu'elles correspondent à l'un ou à l'autre sur base d'un critère de partitionnement. Le même processus est appliqué jusqu'à ce que les éléments se retrouvent sur des feuilles.

Dans la littérature, l'exemple le plus fréquemment utilisé pour expliquer la construction d'un arbre de décision est celui donné par Quinlan, qui est par ailleurs la personne à l'origine de deux algorithmes d'induction d'arbres de décision de référence ID3 et C4.5, dans (Quinlan, 1993). Cet exemple, nommé « weather » a été repris par de nombreux chercheurs, notamment Rakotomalala (Rakotomalala, 2005) et Simon Bernard (Bernard, 2009). Dans ce chapitre, nous n'allons pas utiliser cet exemple mais plutôt un exemple tiré du chapitre 6 du présent mémoire de thèse.

En effet, le Tableau 5.1, présente le dataset que nous avons utilisé pour implémenter la principale contribution de thèse alors que la Figure 5.2 donne l'exemple d'un arbre de décision (DT) qui pourrait être construit à partir de ce dataset. Pour simplifier l'exemple, nous n'avons retenu que les variables qualitatives du dataset.

Tableau 5.1 : Entête du dataset "covid-19 positive-negative"

Gender	Sore throat	Severity	Runny nose	Nasal congestion	Body pain	Difficulty in breathing	Contact with Covid patient	Infected
Male	1	Mild	0	0	1	0	No	0
Male	0	Moderate	1	0	1	0	Not known	1
Female	0	Mild	1	0	0	1	Yes	1
Male	0	Moderate	1	1	1	1	Yes	1
Female	1	Mild	0	0	1	0	Not known	1

Female	1	Mild	0	0	0	0	No	0
Male	1	Moderate	0	1	0	1	Not known	1
Male	0	Moderate	1	1	0	1	Yes	1
Female	0	Mild	1	0	0	0	No	0

L'objectif de cet exemple est de prédire si « oui » ou « non » un patient (une personne) est positif à la covid-19. L'exemple comporte huit variables (caractéristiques) hormis la variable cible et quelques deux mille cinq cents lignes (entrées). Evidemment, dans le cadre de cet exemple, nous n'allons pas exploiter toutes ces lignes. Il faudra ainsi partir de ces huit autres variables pour déterminer si le patient a contracté la covid-19 ou non.

Le principe de construction de l'arbre, énoncé ci-haut, veut que toutes les données (toutes les entrées, lignes) de l'ensemble soient placés dans un seul nœud au départ et qu'une variable de partitionnement soit choisie pour les séparer en différents nœuds fils. La Figure 5.2 illustre l'arbre construit à partir de cet ensemble de données. Sur cette figure, nous pouvons remarquer que la variable de partitionnement de départ choisie est « *Contact_with_covid_patient* ». Le rôle de cette variable est de subdiviser le jeu de données en des sous-groupes plus homogènes. Cette subdivision se fera sur la base des valeurs de cette variable. Nous avons ici trois valeurs : « Yes », « No » et « Unknown » qui permettront de subdiviser l'ensemble de données en trois sous-groupes : ceux qui sont certains d'avoir été en contact avec des patients covid, ceux qui sont certains du contraire et ceux qui ne peuvent ni affirmer ni infirmer qu'ils ont été en contact avec des patients covid.

Par la suite, la croissance de l'arbre se poursuivra par la division des nœuds fils en des nœuds encore plus homogènes sur la base d'une règle de partitionnement et à chaque étape, est calculée une corrélation entre la variable cible et le sous-ensemble de variables qui précède celle présente, sur le parcourt de l'arbre. Cette approche de calcul de corrélation consiste à determiner la corrélation entre la variable cible et chaque variable du sous-ensemble de variables considérées. Cela nous donnera une série de coefficients de corrélation qui reflètent la relation entre la variable cible et chaque variable du sous-ensemble. Ensuite, on calcule la corrélation globale de la feuille, comme une moyenne pondérée des corrélations avec les autres variables.

Dans notre exemple, la prochaine variable de partitionnement est « *Difficulty_in_briething* ». Elle apporte un niveau de tri supplémentaire. Après avoir séparé les entrées en termes de contact avec les patients covid, cette fois, la séparation se fera sur la base d'un symptôme clinique.

Nous pouvons remarquer sur la Figure 5.2 que c'est à ce niveau de tri qu'apparait la première feuille. Il s'agit ici du groupe de personnes ayant été en contact avec des patients covid et qui manifestent des difficultés respiratoires. Ce groupe de personnes est étiqueté « positif » à la covid-19 avec une corrélation de 0.841 avec la variable cible **Infected (**désignant la maladie).

La première feuille de la branche qui regroupe les individus n'ayant pas été en contact avec les patients covid n'apparait qu'après la prise en compte de la troisième variable de partitionnement « *Body_pain* ». Seuls les individus de cette branche qui en plus de ne pas avoir des difficultés respiratoires n'ont pas de douleurs corporelles sont directement étiquetés « négatif » avec une corrélation de 0.126. C'est à ce même niveau (« *Body_pain* ») que la première feuille de la branche qui regroupe les individus ne sachant pas s'ils ont été en contact ou pas avec des patients covid apparait. Les individus de cette branche ayant des difficultés respiratoires et des douleurs corporelles sont étiquetés « positif » avec une corrélation de 0.661.

Dans la suite de l'arbre de décision, nous pouvons observer plusieurs feuilles à différents niveaux de partitionnement. Une feuille peut se définir comme étant un nœud qui n'a pas de fils parce qu'il n'est plus nécessaire de la subdiviser car elle a atteint un niveau d'homogénéité jugé satisfaisant. Mais toutes les feuilles n'ont pas le même niveau d'homogénéité. C'est là qu'intervient la notion de la pureté des feuilles.

La croissance d'un arbre s'arrête lorsque tous ses nœuds sont des feuilles. Une feuille est dite pure lorsque tous les éléments qui la constituent appartiennent à une seule classe, sinon elle est dite impure. Cette notion de la pureté de la feuille est très importante, dans la mesure où elle détermine à la fois la fin de la croissance d'un arbre mais aussi sa capacité en généralisation (Bernard, 2009).

L'un des défis des arbres de décision est le choix de la variable de partitionnement mais aussi la mesure du degré de pureté des feuilles. La variable de partitionnement est choisie au départ pour subdiviser l'ensemble de données mais ce n'est pas le seul endroit où elle est nécessaire. Dans la suite, pour la croissance de l'arbre, il faudra aussi subdiviser les nœuds résultants. Et pour cela il faudra sélectionner les variables de partitionnement à chaque nœud, comme nous l'avons vu dans l'exemple du dataset « covid19 positive-negative ».

C'est une problématique à laquelle se sont intéressés de nombreux chercheurs depuis l'introduction des algorithmes d'induction d'arbre par Breiman en 1984 (Breiman, et al., 1984) et par Quinlan en 1986 (Quinlan, 1986).

2. La phase d'élagage et attribution des conclusions

L'élagage[1] consiste à supprimer, les branches jugées peu représentatives. Il permet de réduire la taille et la complexité afin d'augmenter la capacité en généralisation de l'arbre de décision. Il existe un grand nombre de techniques permettant aux arbres de décision de ne pas sur-apprendre les données d'apprentissage afin de faciliter la généralisation. Toutefois, il faudra noter que les arbres de décision sont réputés pour avoir comme grande faiblesse le manque de généralisation.

Après l'élagage, vient l'étape de l'attribution des conclusions qui consiste à attribuer une conclusion à chaque feuille. Plus la feuille est pure, plus la conclusion est simple à formuler. Dans notre exemple, plus nous avançons vers le bout de l'arbre (du côté opposé à la racine) plus nous trouvons des feuilles pures.

Par exemple, la feuille (Infected, 0.841) de la branche « Yes » issue « *Contact_with_covid_patient* » qui vient juste après « Diff_in_breathing » est moins pur que la feuille (Infected, 0.648) qui vient après « *Nasal_Congestion* », dans la mesure où la probabilité de trouver une entrée étiquetée « *Not Infected* » dans la première feuille est plus importante que dans la deuxième.

En revanche, les feuilles (Not infected, 0.300) et (Infected, 0.512) sont les plus pures de l'arbre parce qu'elles sont au bout de la chaine, elles ont parcouru toutes les caractéristiques et il n'y a pas de risque d'y trouver des entrées avec des étiquettes différentes.

Voici, parmi tant d'autres, quelques-unes des conclusions auxquelles nous pouvons arriver partant de l'exemple sur la Figure 5.2 :

- Les individus ayant été en contact avec les patients covid et qui ont des difficultés respiratoires, mal au corps, des congestions nasales et le nez qui coule sont atteint de la covid-19.
- Les individus qui n'ont pas été en contact avec les patients covid et qui n'ont ni des difficultés respiratoires, ni mal au corps, ni des congestions nasales, ni encore le nez qui coule, mais manifestent une forme de la maladie modérée ou faible, ne sont pas atteint par la covid-19 (ils sont négatifs).

[1] en anglais pruning.

- Les individus qui ne savent pas s'ils ont été en contact avec des patients covid mais ont de difficultés respiratoires, mal au corps, congestions nasales ou qui ont un nez qui coule sont positifs à la covid-19.

Le choix des variables successives de partitionnement s'est fait grâce à l'application du critère de Gini. Notons par ailleurs que le dataset utilisé pour générer cet arbre de décision est celui utilisé dans le chapitre 6 de ce présent mémoire de thèse

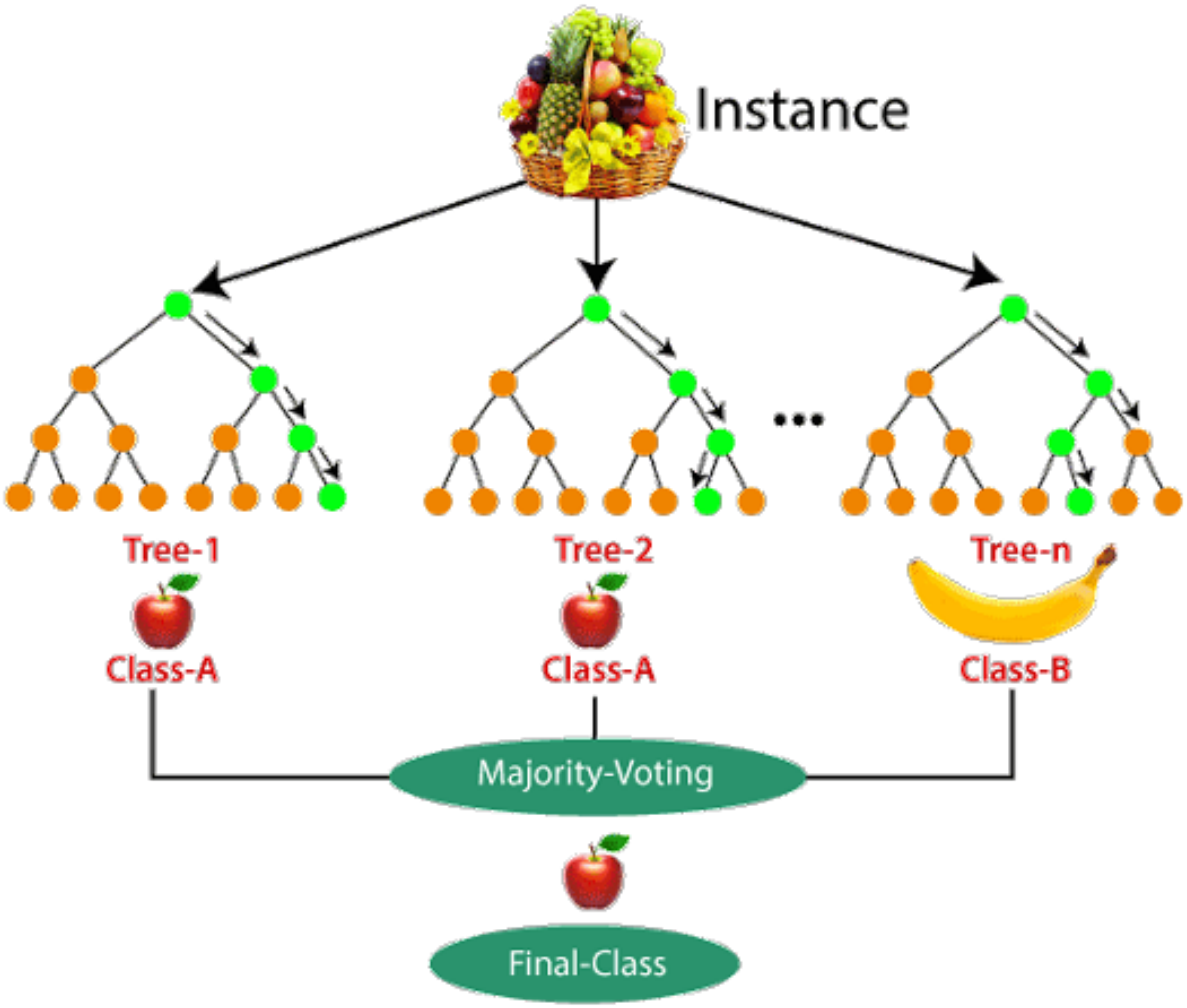


Figure 5.2 : Construction d'un arbre de décision à partir d'un dataset de sur la covid-19

5.3. Les ensembles de classifieurs (EoC)

Bien que les arbres de décision se soient montrés particulièrement efficaces à leur apparition, très vite ils ont affiché quelques limites. Ce qui a poussé les chercheurs à s'intéresser à la possibilité d'associer plusieurs classifieurs pour obtenir des meilleurs résultats. La raison d'être des ensembles de classifieurs peut se résumer en trois points :

- *Le gain de précision :* Il nous semble être la raison la plus intuitive pour l'usage des EoC. En effet, la combinaison de plusieurs classifieurs permet d'obtenir des résultats plus fiables. Il s'agit de comparer les sorties de plusieurs classifieurs pour éviter au maximum le risque de retenir la mauvaise classe (classification) ou valeur (régression). C'est comme si, un malade allait voir trois ou quatre spécialistes pour s'assurer du bon diagnostic, si le même diagnostic est répété par plusieurs médecins alors il est très probablement le bon.

  Dans le cadre des algorithmes d'apprentissage, choisir une étiquète par vote permet de s'assurer que c'est le meilleur choix contrairement à un choix effectué sur la base d'un seul classifieur (Dietterich, 2000). En effet, un algorithme d'apprentissage peut être compris comme un algorithme de recherche d'un classifieur (ou hypothèse). Il arrive des situations où l'espace d'induction de l'algorithme est de loin grand par rapport aux données disponibles en apprentissage. Dans une telle situation, plusieurs classifieurs peuvent se montrer suffisamment performants pour être choisis par l'algorithme, alors que ce dernier ne peut en choisir qu'un seul.

  Il faudra noter que l'efficacité des classifieurs dans cette situation est relative, car elle est liée aux données présentes dans le dataset d'apprentissage. Une petite variation de ces données produira des changements considérables. C'est là que la combinaison des classifieurs s'avère très efficace. Car non seulement elle évite de choisir le mauvais classifieur, mais elle permet aussi d'en créer un autre plus performant que tous les classifieurs présents dans l'espace de recherche. La combinaison des classifieurs permet ainsi de réduire la variance et de gagner en précision. Nous reviendrons sur cette notion de variance dans les lignes qui vont suivre.

- *La complexité du problème à traiter :* Lorsqu'il s'agit d'un problème très complexe, il arrive très souvent qu'aucun classifieur ne soit, à lui seul, capable de le résoudre ou/et d'en saisir le contour. Dans ce cas, l'utilisation d'un seul classifieur produirait un biais

très élevé. Le biais est l'erreur qui correspond en quelque sorte à combien la prédiction passe à côté du point considéré (Chaouche, 2021), autrement dit à quel point notre résultat est « faux ».

Le biais est le résultat d'un manque de relation pertinente entre les entrées et la sortie. Cela est dû aux hypothèses erronées dans l'algorithme d'apprentissage. Cette erreur survient souvent lorsque l'algorithme utilisé est plus simple que le problème à apprendre, car dans ce cas il n'arrivera à rendre compte de toute sa complexité. Pour diminuer le biais, il faudra que la complexité de l'algorithme s'approche de celle du problème. Par exemple le perceptron est un modèle de classification linéaire trop simple pour des problèmes complexes de classification d'images comme CIFAR[2] : il produira un biais très élevé.

La nature de l'erreur dépend du type de problème considéré. Par exemple, dans un problème de classification d'images, l'erreur pourrait être « le % de fois où l'algorithme se trompe en choisissant les classes » ; dans le cadre d'un problème de régression, le biais pourrait être une erreur des moindres carrés, etc. Quoi qu'il en soit, l'erreur totale n'est jamais nulle, ne serait-ce qu'à cause du bruit. Cependant, elle peut être très faible.

C'est là tout l'intérêt des EoC. Lorsque aucun classifieur n'arrive à mieux approximer le problème, il est alors possible de diviser ledit problème en plusieurs sous-problèmes et de les résoudre un par un en utilisant des classifieurs différents. Le résultat final sera donc la combinaison des résultats intermédiaires.

De nombreux travaux sur la résolution de problèmes de classification multi-classes ont été résolus à l'aide d'algorithmes d'apprentissage destinés aux problèmes à deux classes. Pour résoudre un problème de ce type avec des classifieurs exclusivement binaires, c'est-à-dire ne prenant en charge que des problèmes à deux classes, une stratégie souvent adoptée consiste à résoudre séparément plusieurs des sous-problèmes binaires qu'il contient. La décision finale peut alors être prise par combinaison des décisions prises pour les problèmes intermédiaires.

- *Le théorème « No Free Lunch »* : Ce théorème est une des raisons pour lesquelles la combinaison des classifieurs est nécessaire. En effet, ce théorème statue qu'aucun

[2] CIFAR est un jeu de données d'images avec 10 classes d'images, communément utilisé par la communauté pour tester des modèles d'apprentissage.

algorithme (et donc classifieur) ne fonctionne bien pour tous les problèmes. En d'autres termes, si un algorithme fonctionne bien sur un type de problème, il sera moins performant sur d'autres. En définitive, il n'existe pas d'algorithme magique qui serait efficace en toute circonstance. Ce théorème ramené au niveau des classifieurs explique la pertinence des EoC (Chaouche, 2021) et (Wolpert & Macready, 1997).

En somme, nous voyons au travers de ces trois principales raisons que les EoC permettent d'avoir des résultats fiables en s'appuyant sur l'avis de plusieurs experts au lieu d'un seul. Ils permettent aussi d'élargir l'ensemble de solutions possibles en proposant des algorithmes plus complexes qu'il n'est pas possible d'obtenir avec de simples classifieurs. En outre, les EoC sont aussi beaucoup plus performant en généralisation que les classifieurs uniques. Ils sont aussi capables de résoudre des problèmes très complexes, que ne peuvent résoudre des classifieurs uniques. Et ils sont plus génériques que les classifieurs uniques.

Il faudra aussi noter que les EoC, permettent de résoudre une problématique importante des algorithmes d'apprentissage en Machine Learning (ML) : *l'équilibre biais/variance.* Nous définirons la variance dans les lignes qui suivent. La problématique de l'équilibre biais/variance est centrale en machine learning (Cayla, 2021). Trouver cet équilibre est l'ingrédient principal d'un bon modèle ML. Cependant, il n'est pas si simple à trouver. Bien qu'il existe des techniques d'ajustement de ces deux indicateurs, il n'existe pas de recette miracle pouvant donner un équilibre parfait. Pour bien comprendre la difficulté à trouver cet équilibre, prenons un exemple.

Nous avons ici un exemple de données représentées sur la Figure 5.3.

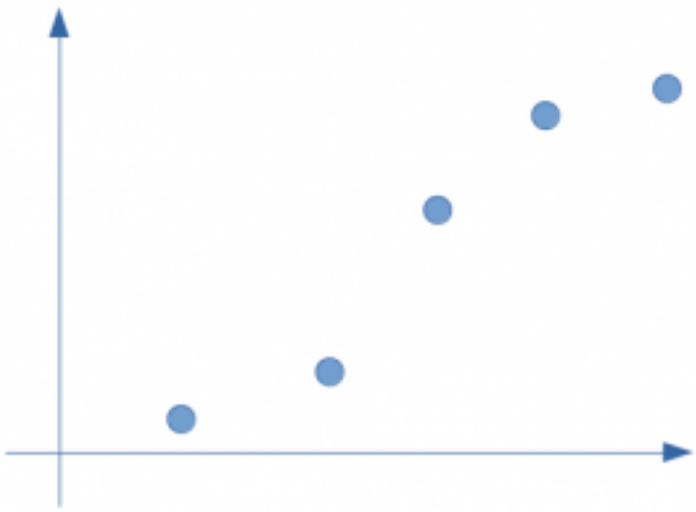


Figure 5.3 : Equilibre biais/variance - exemple de données (Cayla, 2021)

A la vue de ces données, une régression linéaire pourrait suffire pour résoudre ce problème. Après modélisation et calcul de l'erreur, nous pouvons aboutir au résultat présenté à la Figure 5.4.

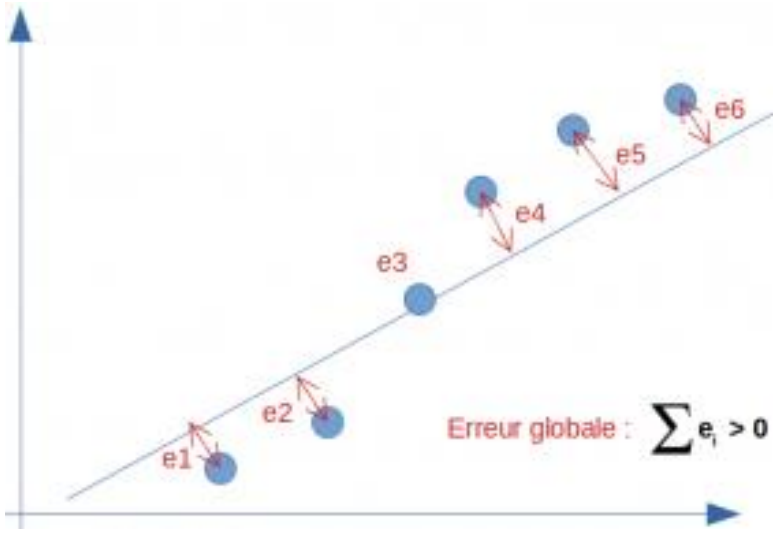


Figure 5.4 : Equilibre biais/variance en cas de régression linéaire, (Cayla, 2021)

Nous remarquons sur la Figure 5.4 qu'avec une régression linéaire l'erreur ou le biais est supérieur à zéro voire largement supérieur car la droite ne passe pas par tous les points. Il est donc possible d'améliorer notre modèle afin d'obtenir un biais minimum. C'est ce que nous allons faire et le résultat est illustré par la Figure 5.5.

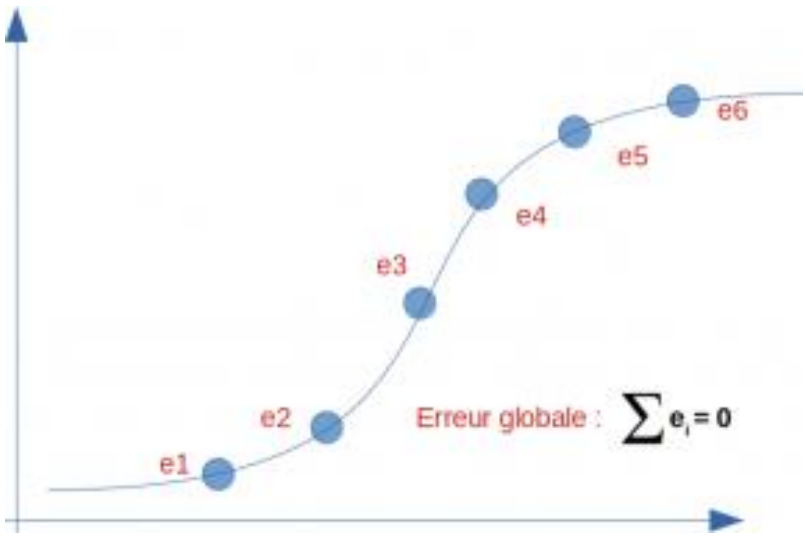


Figure 5.5 : Equilibre biais/variance - amélioration de la courbe, (Cayla, 2021)

Dans ce nouveau cas de figure, la courbe passe par tous les points et donc le biais est de presque zéro. Pour évaluer la variance, nous allons tester notre modèle avec un jeu de données qu'il ne connait pas. Le nouveau jeu de données est indiqué par les points rouges sur la Figure 5.6.

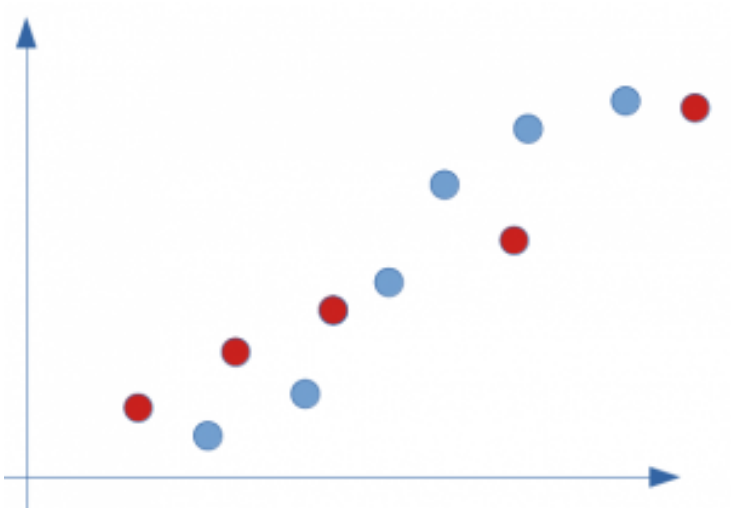


Figure 5.6 : Equilibre biais/variance - nouveau jeu de données pour mesurer la variance, (Cayla, 2021)

Si nous appliquons notre modèle sur le nouveau jeu de données, nous remarquerons, sur la Figure 5.7, que l'erreur (le biais) n'est plus aussi minime qu'elle l'était sur les données d'entrainement (Figure 5.5).

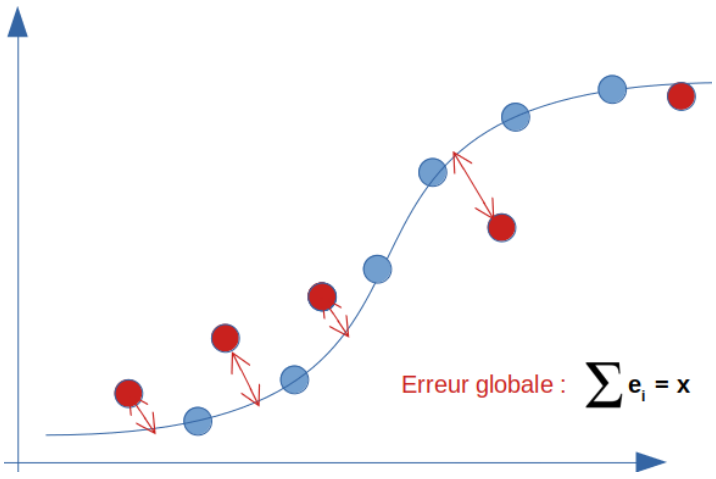


Figure 5.7 : Equilibre biais/variance - application du modèle sur le nouveau jeu de données, (Cayla, 2021)

Pour un nouveau jeu des données, notre modèle perd sa capacité à minimiser l'erreur. Il nous produit à nouveau une erreur quasi-maximal. Ceci s'explique par le fait qu'il était très collé aux données d'entrainement. Ce phénomène s'appelle en machine learning le sur-apprentissage

(overfitting en anglais). Nous avons ainsi une variance élevée. La variance est l'erreur due à la sensibilité aux petites fluctuations de l'échantillon d'apprentissage (Cayla, 2021).

En général, minimiser le biais revient à maximiser la variance et inversement. Tout l'enjeu du data scientistes consiste à trouver l'équilibre biais/variance. Autrement dit un taux de biais acceptable. C'est aussi en cela que les EoC sont plus intéressants que les classifieurs uniques.

D'une part, la capacité des EoC à proposer des modèles qui représentent au mieux le problème à résoudre leur confère de pouvoir minimiser le biais. D'autre part, leur forte capacité en généralisation, leur permet de fournir des algorithmes capables de comprendre et résoudre de nouveaux problèmes, minimisant ainsi la variance. Nous avons ainsi avec les EoC des modèles et algorithmes dont l'équilibre biais/variance (ou variance/biais) est de loin meilleur que celle des classifieurs uniques.

Pour finir cette introduction sur les raisons d'être des EoC, soulignons le fait que toutes ces qualités et caractéristiques ne peuvent réellement exister que s'il y ait de la diversité au sein de l'ensemble des classifieurs (Kuncheva & Whitaker, 2003), (Kuncheva, 2003) et (Kuncheva, 2014). Ceci renvoie au principe de la sagesse de la foule. La sagesse de la foule est un principe qui se base sur l'idée que deux valent mieux qu'un.

Le principe de la sagesse de la foule stipule qu'une décision prise par votes d'un ensemble d'individus amateurs est meilleure que celle prise par un seul expert (Frey & Van de Rijt, 2021). Cependant, pour que cette foule soit plus compétente que l'expert, elle doit vérifier que les individus qui la constituent soient indépendants les uns des autres. Sans cette propriété, aucune des qualités citées ci-haut n'est possible pour un EoC. La diversité exprime la différence des classifieurs au sein d'un EoC.

Il ne s'agit pas ici d'une différence de la nature des classifieurs, même si parfois c'est le cas (nous le verrons dans la section suivante), mais plutôt de leurs prédictions. Il va de soi, que si tous les classifieurs produisaient tous les mêmes prédictions, cela n'aurait aucun intérêt. La qualité d'un EoC repose ainsi sur la diversité de ses classifieurs. Dans cet objectif, il existe plusieurs manières de construire un EoC. Nous allons nous y atteler dans la section suivante.

5.3.1. Architectures et types d'ensembles de classifieurs

La construction des EoC a fait l'objet de plusieurs travaux de recherche. Au fil des années, la communauté scientifique a tenté de regrouper toutes ces constructions en différentes catégories

(Heutte, 1994), (Moobedl, 1996), (Rahman & Fairhurst, 1999), (Dietterich, 2000), (Amit, et al., 2000) (Roli, et al., 2001), (Rahman & Fairhurst, 2003), (Kuncheva, 2014) et (Zouari, 2004).

Dans cette thèse, nous utiliserons le terme *architecture* pour désigner le schéma de combinaison des classifieurs et celui de *type* pour désigner la nature des classifieurs utilisés.

5.3.1.1. Architectures des EoC

L'architecture des EoC, est le plus souvent en combinaison. Trois combinaisons sont à distinguer : série, parallèle et hybride.

- ✓ EoC en combinaison série

Cette architecture implique que les classifieurs seront dépendants les uns des autres. La décision qui sera prise par un classifieur tiendra compte de celle du classifieur précédent (Rahman & Fairhurst, 2003), (HEUTTE, 2000). Pour ce faire la combinaison s'organisera en niveaux successifs de décision permettant de réduire progressivement le nombre de classes possibles. A chaque niveau un seul classifieur décide en tenant compte de la prédiction donnée par le classifieur en amont.

Cette architecture a pour avantage la réduction de l'ambigüité au travers du filtrage progressif des décisions. Cependant, elle présente quelques inconvénients tels que : (1) La sensibilité à l'ordre dans lequel sont placés les classifieurs. Il suffit de modifier la position d'un classifieur pour obtenir un résultat différent. (2) Elle suppose une connaissance préalable de chaque classifieur. Parce que les prédictions des classifieurs dépendent de celle des autres, il est important de bien connaitre chaque classifieur afin de construire un bon EoC. (3) Difficile à optimiser. Le caractère série qui rend chaque classifieur dépendant des classifieurs précédents rend difficile la manipulation du système EoC dans son ensemble (Kittler, et al., 2003) et (Kittler, 2004) .

- ✓ EoC en combinaison Pallèle

Dans cette architecture, les classifieurs sont totalement indépendants. Chaque classifieurs effectue sa prédiction sans tenir compte de celle des autres classifieurs de l'EoC, ensuite toutes les réponses sont fusionnées pour produire une réponse commune. La réponse commune est produite sur base d'un processus qui cherche le consensus. Ce processus pourrait être un vote à la majorité.

Cette architecture présente l'inconvénient d'être gourmande (consomme beaucoup de mémoire vive) à cause de l'activation de tous les classifieurs en même temps. Il est par ailleurs, facile à

mettre en œuvre et ne nécessite pas une re-paramétrisation de tous les classifieurs en cas de modification d'un seul.

La combinaison parallèle est l'architecture qui suscite un grand intérêt de la communauté scientifique (Kittler & Roli, 2000), (Roli & Kittler, 2002), (Windeatt & Roli, 2003), (Oza, et al., 2005), (Haindl, et al., 2007) et (Benediktsson, et al., 2009), principalement à cause de la facilité de sa mise en œuvre, mais aussi parce que les systèmes multi-classifieurs qui utilisent, même partiellement, la combinaison série sont difficiles à optimiser.

- ✓ EoC en combinaison Hybride

Cette architecture a été conçue dans le but de bénéficier des avantages des deux précédentes. Comme son nom l'indique, elle consiste en un mélange des combinaisons série et parallèle. Certains classifieurs seront organisés en série avant de laisser place à une organisation en parallèle ou inversement. Les méthodes appartenant à cette catégorie sont généralement conçues pour des applications spécifiques, comme c'est le cas par exemple de la méthode proposée par Kim et al. Dans (Kim, et al., 2000) pour la reconnaissance des mots cursifs anglais extraits de chèques bancaires.

Cette architecture présente l'avantage de proposer de nombreux schémas de combinaison pour tirer parti au mieux des données, cependant son gros désavantage est qu'elle est souvent très dépendante des données à traiter (il y a donc un problème de généricité) en plus d'être très complexe à optimiser.

5.3.1.2. Types d'EoC

Nous allons ici nous intéresser à la nature des classifieurs qui composent l'ensemble. De ce point de vue, nous distinguons deux types d'EoC : Hétérogène et Homogène.

Les EoC Hétérogènes sont ceux qui sont composés des classifieurs de type différents (par exemple un perceptron et un k-NN). L'étude de Kittler et al. dans (Kittler, et al., 1998) où les auteurs s'intéressent à la combinaison de trois classifieurs de types différents, (Bayésien, un Réseau de Neurones), et un Modèle de Markov Cachés (HMM)) pour la reconnaissance d'écriture manuscrite en est un exemple pratique.

Les EoC Homogènes sont ceux qui sont composés des classifieurs de même nature. C'est ce type d'EoC qui va particulièrement nous intéresser dans la suite de ce chapitre car la méthode Random Forest en fait partie. En effet, comme nous le verrons dans la suite, Random Forest est un EoC qui ne rassemble que des arbres de décision.

5.3.2. Principes d'induction d'ensembles de classifieurs

Tout l'enjeu de l'induction des classifieurs est de produire, partant de plusieurs classifieurs élémentaires moyennement performants, un super-classifieur (multi-classifieur) efficace qui minimise la variance sans augmenter le biais. Pour y arriver, il existe plusieurs approches possibles que nous pouvons retrouver dans la littérature (Kuncheva, 2014), (Dietterich, 2000) , (Banfield, et al., 2004) et (Banfield, et al., 2006).

L'une de ces approches consiste en la manipulation des données de l'ensemble des données d'apprentissage. Cette approche consiste à subdiviser l'ensemble des données d'apprentissage en plusieurs sous-ensembles. Par la suite, chaque classifieur se verra affecter un sous-ensemble des données qu'il manipulera sans être au courant du reste de l'ensemble de données. Ce procédé permet d'aboutir à des prédictions différentes.

Une autre approche consiste en la manipulation des variables. Le procédé reste le même que celui décrit précédemment. L'ensemble des variables est subdivisé en plusieurs sous-ensembles, ensuite affectés aux classifieurs dans le but de produire de la diversité dans le comportement de ceux-ci. Une troisième approche est de se focaliser sur la manipulation des paramètres. Il s'agit ici d'induire des classifieurs élémentaires avec des paramètres différents. Ces paramètres sont souvent choisis de manière aléatoire.

La liste de ces approches n'est pas exhaustive, elle couvre cependant la majorité des principales méthodes d'induction d'ensembles des classifieurs que nous essayons de détailler dans la suite de cette section.

5.3.2.1. Bagging

Ce principe d'induction d'ensemble des classifieurs fait partie de l'approche qui consiste à manipuler les données. Au départ conçu en 1993 comme une simple méthode statistique (Efron & Tibshirani, 1993), le Bagging a été popularité par Breiman qui l'a repris quelques années plus tard pour l'appliquer aux arbres de décision (Breiman, 1996).

Le Bagging pour boostrapp aggregating est un principe de sous-échantillonnage avec remise. Cela signifie qu'une partie des données de l'ensemble d'apprentissage est tirée au hasard et affectée au classifieur ensuite rendue dans l'ensemble pour un nouveau tirage destiné à un nouveau classifieur. Ainsi chaque classifieur sera entrainé sur un ensemble d'apprentissage différent.

Ce principe de Bagging peut être appliqué à tout type d'algorithme d'apprentissage. Cependant il est plus adapté à ceux qui utilisent des classifieurs instables. Car la force du bagging est de réduire l'instabilité (El Alaoui, 2014). Il est donc plus adapté aux algorithmes à forte variance. Ceci explique pourquoi Breiman en 1996 (Breiman, 1996) l'applique aux arbres de décision. En effet, dans le cas précis des arbres de décision, nous entendons par instabilité qu'un petit changement dans la base d'apprentissage provoque un changement important dans la structure de l'arbre et donc dans ses performances en généralisation. Réduire l'instabilité permet alors dans ce cas de fiabiliser les prédictions et d'améliorer les performances en généralisation. Or, les arbres de décision sont des classifieurs très instables. D'où l'importance d'utiliser le Bagging.

D'ailleurs, Skurichina et Grandvalet dans (Skurichina, 2001) et (Grandvalet, 2004) démontrent que le Bagging ne permet pas d'améliorer les performances d'un classifieur stable tel que le Kpp, K plus proche voisin. Considérant ceci, l'instabilité des arbres de décision est un avantage pour un EoC car les performances de l'ensemble peuvent être améliorées avec l'utilisation du Bagging.

Dans le contexte de Bagging, le résultat final est obtenu soit par la moyenne des résultats des classifieurs, lorsqu'il s'agit d'une régression, soit par vote majoritaire lorsqu'il s'agit d'une classification.

5.3.2.2. Boosting

Ce principe se base sur des pondérations données aux classifieurs et aux données d'apprentissage. Il s'agit à l'itération n de concentrer l'apprentissage du classifieur h_n sur les erreurs des classifieurs $h_{n-1}, h_{n-2} \ldots h_1$, obtenus aux itérations précédentes. Cet objectif est atteint à l'aide d'une pondération des données d'apprentissage.

À la première itération, toutes les données de l'ensemble d'apprentissage se voient attribuer un poids identique, tel que la somme de ces poids soit égale à 1. Puis un classifieur dit *"faible"* est construit sur cet ensemble, pour lequel le taux d'erreur en apprentissage est non nul, mais nécessairement inférieur à 0.50. La pondération est ensuite mise à jour sur la base de ces erreurs de prédiction, de sorte à augmenter le poids des données d'apprentissage qui ont été mal classées par ce classifieur, tout en diminuant simultanément les poids des données bien classées. Ainsi, on spécialise progressivement les classifieurs pour qu'ils se concentrent sur l'apprentissage des données précédemment mal classées. C'est ainsi que le résultat est amélioré progressivement.

Une des conditions pour que ce processus fonctionne est que les classifieurs utilisés comme classifieurs élémentaires soient "faibles". Dans ce cadre, un classifieur est dit faible, lorsqu'il est au moins aussi bon qu'une prédiction complètement aléatoire. C'est à dire que le taux de bonnes prédictions du classifieur est supérieur ou égal à 50%. L'algorithme d'apprentissage le plus connu utilisant ce principe est Adaboost (Schapire, 1990).

5.3.2.3. Random Subspaces

Le principe des Random Subspaces (noté RSM pour Random Subspace Method) est assez proche dans l'idée au Bagging (Ho, 1995), (Ho, 1998) et (Brill, et al., 2003). Cette fois, il ne s'agit plus cependant, de manipuler les données d'apprentissage, mais plutôt de manipuler les caractéristiques (variables). Le principe de base est d'entrainer chaque classifieur élémentaire sur un sous-espace aléatoire de l'espace de description.

Chacun de ces sous-espaces aléatoires est de même dimension P, avec $P < M$ où M est la dimension de l'espace de description original.

Plus concrètement, pour l'induction d'un classifieur élémentaire h_k, il s'agit :

1. D'effectuer un tirage aléatoire sans remise de P caractéristique parmi les M caractéristiques disponibles ;
2. De projeter toutes les données d'apprentissage dans ce nouveau sous-espace des caractéristiques ;
3. D'entrainer le classifieur h_k sur ces projections des données d'apprentissage.

Au sujet de ce procédé (RSM), Ho démontre dans (Ho, 1998) qu'il est plus efficace lorsque l'espace de description contient des informations redondantes dispersées dans l'ensemble des caractéristiques plutôt que concentré dans un sous-ensemble.

Le RSM a aussi un gros avantage de réduire considérablement le coût computationnel. Ceci s'explique simplement par le fait qu'il permet de réduire l'espace de description des données d'apprentissage utilisé par chaque classifieur. C'est un avantage que n'offre pas d'autres principes d'induction des EoC. Dans la littérature, beaucoup de travaux ont appliqué ce principe aux arbres de décision : (Ho, 1992), (Ho, 1995), (Banfield, et al., 2006), (Latinne, et al., 2000) et (Latinne, et al., 2002). Son succès dans le monde des arbres de décision peut s'expliquer par le fait que les arbres de décision nécessitent un parcours répété de l'espace de description des données, avec le RSM, il y a un gain considérable du temps de calcul.

5.4. Forêts aléatoires et Algorithmes d'induction

La section précédente a posé les bases de la formalisation de la méthode de Forêt Aléatoire (Random Forest). En effet, les Forêts Aléatoires font partie des méthodes d'ensemble des classifieurs. Dans cette section, nous allons poursuivre ce formalisme en allant de plus en plus dans les détails de son fonctionnement.

5.4.1. Forêts aléatoires

Les années 90 ont vu apparaitre les premiers travaux sur les combinaisons des arbres de décision. Citons pour exemples les travaux de Shlien en 1990 (Shlien, 1990), Dietterich en 1991 (Dietterich & G.Bakiri, 1991), Ho en 1992 (Ho, 1992), Ho en 1995 (Ho, 1995), Breiman en 1996 (Breiman, 1996), Amit et Geman la même année (Amit & Geman, 1996) et de Ho en 1998 (Ho, 1998). Bien que la liste de travaux que nous venons de donner ne soit pas exhaustive, elle résume tout de même assez bien la démarche des chercheurs qui ont contribué à la formalisation de la méthode Forêt Aléatoire, ainsi que les principes sur lesquels est fondée cette méthode.

En effet, en 1990, Shlien (Shlien, 1990), propose d'utiliser un ensemble d'arbres de décision (une forêt d'arbres) pour la reconnaissance de caractères manuscrits. En 1991, un premier principe qui permet d'améliorer la combinaison d'arbres de décision est proposé par Dietterich. Il s'agit de la méthode de Error Correcting Output Codes (ECOC). Cette méthode permet d'améliorer les performances des multi-classifieurs binaires. En 1992 Ho met en évidence le gain de performance qui permet d'obtenir des classifieurs qui utilisent plusieurs arbres de décision en combinaison parallèle. Par la suite, en 1995, elle devient la première à utiliser le terme *Decision Forest* pour désigner les ensembles d'arbres de décision. Il s'agit en effet, du premier nom de Random Forest que nous connaissons aujourd'hui.

Cependant, il faudra attendre 1996 pour voir apparaitre un des principes de base du Random Forest : *Bagging,* que nous avons exposé dans la section précédente. Proposé par Breiman, le *Bagging* est un principe d'apprentissage d'ensemble des classifieurs qui peut être utilisé avec tout type de classifieur élémentaire, mais dont l'efficacité est démontrée principalement dans le cadre de combinaison d'arbres de décision. La même année, le deuxième principe de base de Random Forest est introduit. C'est une méthode de randomisation des arbres de décision : Random Feature Selection.

Nous pouvons ainsi retenir que les deux principes de base d'induction des forêts aléatoires sont : le *Bagging* et le *Random Feature Selection*. Nous constatons ainsi que Leo Breiman n'est pas

le seul fondateur de Random Forest, bien que ce soit lui qui a proposé le terme Random Forest et défini le cadre formel de celui-ci, dans son article publié en 2001 (Breiman, 2001).

Dans cet article fondateur du Random Forest (Breiman, 2001), Leo Breiman définit une forêt aléatoire comme un classifieur constitué d'un ensemble de classifieurs élémentaires de type arbres de décision, notés :

$$\{ \boldsymbol{h(x, \Theta_k), k\ variant\ de\ 1\ à\ L} \}$$

Où $\{\Theta_k\}$ est une famille de vecteurs aléatoires indépendants et identiquement distribués, et au sein duquel chaque arbre participe au vote de la classe la plus populaire pour une donnée d'entrée x.

Cette définition donne les deux éléments clés qui définissent les forêts aléatoires :

1. Les forêts aléatoires sont basées sur des ensembles d'arbres de décision.
2. Une certaine "quantité" d'aléatoire est introduite dans le processus d'induction.

Cependant, il y a un autre élément que nous pouvons tirer en lisant attentivement la définition de Θ_k, une famille de vecteurs aléatoires indépendants et identiquement distribués. Ces deux termes (indépendants, identiquement distribués) introduisent la formalisation de la randomisation des algorithmes d'induction des forêts aléatoires. Cela signifie que chaque arbre de la forêt est induit de manière indépendante des autres arbres déjà ajoutés d'une part. D'autre part, que le principe de randomisation utilisé doit être identique pour tous les arbres de la forêt. Les résultats théoriques en grande partie démontrés par Breiman sur la convergence (Breiman, 2001) et la consistance des forêts (Breiman, 2004), sont fortement basés sur cette hypothèse d'indépendance et de distribution homogène.

Breiman se base sur cette hypothèse pour démontrer la convergence de l'erreur en généralisation pour un nombre croissant d'arbres de décision. En effet, les arbres de décision ont pour grande faiblesse, leur faible capacité en généralisation (overfitting ou sur-apprentissage), Breiman trouve ici le moyen de gommer cette faiblesse. Il démontre donc qu'un nombre croissant d'arbre diminue le taux d'erreur en généralisation. Ce qu'il nomme la convergence de l'erreur en généralisation. Breiman introduit aussi les propriétés de force et de corrélation.

Dans (Breiman, 2001), Breiman démontre la propriété importante de convergence des forêts aléatoires. Plus précisément, il démontre que le taux d'erreur en généralisation, d'une forêt aléatoire converge nécessairement vers une valeur limite, quand le nombre d'arbres qui la composent augmente. Concrètement, cela signifie une chose importante : avec un nombre suffisamment élevé d'arbres, une forêt aléatoire évite le sur-apprentissage. Néanmoins, cette croissance du nombre d'arbres est soumise à deux propriétés intrinsèques des forêts :

- La force : mesure la fiabilité de la forêt.
- La corrélation : mesure le taux de corrélation des classifieurs élémentaires entre eux.

Après une longue démonstration mathématique de ces deux propriétés, Breiman en arrive à la conclusion suivante : *le taux d'erreur en généralisation est d'autant plus faible que l'ensemble des classifieurs élémentaires est fiable,* ce qui naturellement est évident. Il démontre également que *le taux d'erreur en généralisation est encore beaucoup plus faible lorsque les classifieurs sont décorrélés en termes de prédictions.* Tout l'enjeu de l'induction de forêts aléatoires réside donc dans la production d'un jeu d'arbres, les plus performants possibles individuellement et les moins corrélés possibles dans leurs prédictions.

Bien que Breiman ait démontré l'efficacité de la croissance du nombre d'arbres sur la diminution du taux d'erreur en généralisation, la borne de l'erreur en généralisation semble peu précise ainsi que la limite du nombre d'arbre. Jusqu'à ce jour, à notre connaissance, très peu de travaux ce sont penchés sur cette question. Voilà une première problématique des forêts aléatoires qu'il nous semble intéressante d'étudier.

5.4.2. Algorithmes d'induction

La méthode Random Forests a eu au fil des années plusieurs algorithmes proposés par des chercheurs successifs. En ce qui nous concerne, nous nous intéresserons à l'algorithme originel, celui proposé par Leo Breiman dans (Breiman, 2001) Forest RI, (RI pour Random Input), pour la simple raison que c'est l'algorithme par défaut de RF classique et par conséquent l'algorithme le plus utilisé. Néanmoins, avant d'entrer dans les détails du fonctionnement du Forest RI, nous devons au préalable présenter un principe important : Random Feature Selection. Ce principe découle sur l'algorithme Random Tree utilisé par Forest RI.

5.4.2.1. Random Feature Selection

Random Feature Selection a pour objectif principal d'améliorer la décorrélation des arbres de la forêt. Comme dit précédemment, il s'agit là d'une condition pour avoir une forêt de bonne qualité.

Le principe de Random Feature Selection consiste à introduire l'aléatoire dans le choix des règles de partitionnement à chaque nœud des arbres, de sorte que chaque règle ne soit plus choisie à partir de l'ensemble des caractéristiques disponibles, mais plutôt à partir d'un sous-ensemble de ces caractéristiques. Il s'agit plus précisément de sélectionner par tirage aléatoire sans remise d'un nombre K de caractéristiques et de choisir la meilleure des règles possibles utilisant ces K caractéristiques uniquement (Dubey, 2018). L'algorithme d'induction d'un arbre de décision qui intègre ce procédé est généralement appelé Random Tree que nous présenteroons dans la suite.

Outre le fait de produire des arbres décorrélés, ce principe permet aussi de réduire le nombre des calculs à effectuer pour le choix de la règle de partitionnement à chaque nœud de l'arbre. En effet, il faudra noter que ce principe de randomisation des arbres de décision a initialement été introduit dans le cadre des problèmes de reconnaissance d'écriture manuscrite, par Amit et Geman dans (Amit & Geman, 1996).

La représentation des données utilisées dans ces travaux implique que le nombre des règles de partitionnement possibles est trop important pour qu'elles soient toutes testées à chaque noeud. L'idée d'Amit et Geman a donc été de choisir un sous ensemble aléatoire de ces règles et de choisir parmi elles la meilleure afin de réduire les calculs. Breiman reprend cette idée de *Random Feature Selection* puis l'introduit dans Random Forest afin de répondre principalement à un besoin de réduction de la complexité de l'induction des arbres de décision.

5.4.2.2. Forest RI

En plus d'être chronologiquement le plus ancien des algorithmes des forêts aléatoires, Forest RI est le premier à avoir été implémenté par Leo Breiman et Adèle Cutler en langage Fortran 77 (Breiman & Cutler, 2007). Les deux chercheurs ont déposé leur implémentation sous le nom de *Random Forest*. Ceci explique que les logiciels utilisant Random Forests, recourent souvent à une implémentation de Forest RI.

Cet algorithme s'appuie sur deux principes que nous avons déjà présentés : le Bagging et le Random Feature Selection. Il est constitué de deux principaux paramètres :

1. Le nombre des caractéristiques choisies aléatoirement à chaque nœud des arbres : Il est noté par *K* dans l'algorithme et par *mtry* dans le package Fortran de Breiman (Breiman & Cutler, 2007). Il varie entre 1 et p., p étant le nombre total de toutes les variables de la base d'apprentissage. Sa valeur par défaut est de $\sqrt{\boldsymbol{p}}$ pour la classification, et de $\frac{\boldsymbol{p}}{\boldsymbol{3}}$ pour la régression. Ceci dit, ces valeurs par défaut ne donnent pas toujours des résultats optimaux. L'optimisation des résultats est souvent atteinte par réglage à tâtons.
2. Le nombre d'arbres dans la forêt : Là aussi, des chiffres empiriques selon les types de problèmes posés n'existent pas à notre connaissance. L'utilisateur est soumis à faire des tests successifs jusqu'à obtenir des résultats satisfaisants. Dans les algorithmes 1 et 2 qui suivent, ce nombre est noté *L* et *ntree* dans le package Fortran de Breiman (Breiman & Cutler, 2007).

Nous présentons dans Algorithme 1 et Algorithme 2, les étapes de Forest RI et Random RI, respectivement.

Algorithme 1 : Forest RI

Fonction Forest_RI (T, L, K)

Entrée : T, l'ensemble d'apprentissage

Entrée : L, le nombre d'arbres dans la forêt

Entrée : K, le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud

Sortie : foret, l'ensemble des arbres qui composent la forêt construite

1. Pour chaque arbre i **de** 1 **à** L **faire :**

a. Tl ← ensemble bootstrap, dont les données sont tirées aléatoirement (avec remise) de T

b. Créer un arbre vide, i.e. composé de sa racine uniquement

c. arbre.racine ← RndTree(arbre.racine, Tl, K)

d. Ajouter l'arbre à la forêt : foret ← foret ∪ arbre

2. Retourner la forêt construite

Explication de Forest RI : Permet de rassembler l'ensemble des composantes de la forêt aléatoire. Avec *L* le nombre des estimateurs (arbre) à utiliser dans la construction de la forêt, pour chaque arbre nous faisons appel à la méthode de construction d'un arbre *RnTree* afin de construire l'arbre correspondant, en utilisant :

- Des données tirées aléatoirement à partir d'une base de données avec remise Tl
- Des variables ou features sélectionnées aléatoirement K

A chaque itération, un arbre est généré et les aléas utilisés par Bootstrap sur le jeu des données, permettent de supposer l'indépendance d'opinions de ces estimateurs. Enfin ces arbres sont regroupés pour constituer la forêt dite aléatoire du fait de la méthode de construction des arbres.

Pour obtenir le resultat de cette fonction, il suffit de l'appeler comme suit : **Forest_RI(T, L, K)**

Algorithme 2 : Random Tree

Fonction RndTree(n, T, K)
Entrée : n, le noeud courant
Entrée : T, l'ensemble des données associées au nœud n
Entrée : K, le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud
Sortie : n, le même nœud, modifié par la procédure
1. Si n n'est pas une feuille alors :
a. C ← K caractéristiques choisies aléatoirement
b. Pour chaque A appartenant à C **faire :**
i. Appliquer la procédure CART pour créer et évaluer (critère de Gini) le partitionnement produit par A, en fonction de T
ii. partition ← partition qui optimise le critère Gini
iii. n.ajouterFils(partition) #ajoute un nouveau nœud à l'arbre de décision en tant que fils du nœud courant n.
iv. Pour chaque fils appartenant à n.noeudFils **faire :**
1. fils ← RndTree(fils, fils.donnes, K)
2. Retourner n

Explication de Random Tree : Elle permet la mise en place d'un arbre à travers la méthode de partitionnement sur la base des variables aléatoirement choisies K. A partir d'un nœud *n* pour construire le nœud fils ou potentiellement une feuille, nous appliquons pour chaque variable A ∈ C le critère de Gini afin de prendre la partition qui minimise le désordre ou qui permet de bien prédire la valeur réelle. Ce critère pour rappel, est utilisé dans les arbres de décision pour mesurer l'homogénéité d'un nœud et trouver les questions qui divisent le mieux les données en sous-groupes homogènes. CART est un algorithme itératif qui utilise une méthode de division récursive pour construire un arbre de décision. À chaque étape, l'algorithme sélectionne la caractéristique qui sépare le mieux les données en fonction d'un critère de qualité de division, souvent la réduction de l'impureté ou de l'erreur de classification. Ce processus est ensuite utilisé sur chaque nœud pour construire l'arbre de manière récursive.

5.5. Conclusion

Somme toute, ce chapitre a eu pour objectif une meilleure compréhension de la méthode Random Forests grâce à une formalisation de son fonctionnement. Les forêts aléatoires étant des ensembles des classifieurs de types arbres de décision, ce chapitre de formalisation s'est en premier lieu intéressé au fonctionnement de ces derniers.

Nous avons vu dans ce chapitre que les arbres de décision se forment au travers d'un processus en deux phases : la phase d'expansion et la phase d'élagage.

La phase d'expansion consiste à construire la structure hiérarchique de l'arbre suivant un processus récursif de division de l'espace de données en des nœuds de plus en plus homogène. Ce processus consiste à mettre initialement toutes les observations dans un seul nœud appelé nœud racine et par la suite, le nœud racine sera étendu en plusieurs nœuds fils sur la base des variables qu'il contient. Après vient la répartition des données dans les nœuds fils sur la base **d'une règle de partitionnement**. Ce processus ainsi décrit sera répété sur chaque nœud fils jusqu'à l'obtention des feuilles.

La phase d'élagage, quant à elle, consiste à supprimer les branches jugées peu représentatives. Le but étant d'augmenter la capacité en généralisation de l'arbre. Le grand défi dans la construction des arbres de décision est le choix de la règle de partitionnement.

Ce défi est, au moins en partie, relevé dans la construction de Random Forests. En effet, l'algorithme originel de Random Forests que nous avons repris dans la section 4 est *Forest RI.* Celui-ci utilise l'algorithme Random Tree qui contribue à répondre à ce défi du choix de la règle de partitionnement. Cette contribution consiste en la diminution des calculs à effectuer pour le choix de la règle de partitionnement. Ce défi étant résolu, il n'en est pas de même en ce qui concerne la mise en œuvre de Random Forests.

Nous avons vu dans la section 4 que l'algorithme originel de Random Forests, *Forest RI,* possède deux principaux paramètres : K, le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud et L le nombre d'arbres dans la forêt. La mise en œuvre de cet algorithme pose problème dans le choix de ces deux paramètres, comme a pu aussi le souligner (Bernard, 2009). En effet, les utilisateurs de Random Forests ne disposent pas d'indications précises pour vérifier les valeurs que peuvent prendre ces deux paramètres pour optimiser le fonctionnement de Random Forests alors que nous savons que ces paramètres sont importants pour le fonctionnement de l'algorithme. Ainsi les utilisateurs évoluent à tâtons sans avoir des

indications ni sur la valeur maximale ni minimale. Cette difficulté est probablement due au fait que les valeurs de ces deux paramètres sont souvent très corrélées aux données.

Dans cette thèse, nous nous proposons de contribuer non pas à résoudre la problématique des paramètres *K* et *L* qui sont beaucoup trop liés aux données mais d'introduire des mécanismes, dans le cœur même de l'algorithme Forest RI, qui permettront d'améliorer les performances de Random Forests dans l'analyse prédictive épidémiologique. Nous nous donnons comme objectif la limitation de la propagation des virus. Le modèle RF qui implémentera notre algorithme devrait être en mesure des produire des résultats plus satisfaisants que le RF classique qui implémente Forest RI en ce qui concerne la limitation de l'expansion d'un virus. Ceci sera l'objet du chapitre 6.

Chapitre 6 : Enrichissement du modèle et Expérimentation

6.1. Introduction

Le mot *épidémie* quoi qu'intuitivement compréhensible par tous, peut parfois être complexe à définir. Pour aider la communauté des chercheurs à résoudre cette problématique (O'Neil & Naumova, 2007), ont publié un travail de recherche qui élucide les différentes définitions de ce terme. Citons ici trois définitions tirées de (O'Neil & Naumova, 2007).

(O'Neil & Naumova, 2007) définissent une épidémie comme « toute augmentation de l'incidence d'une maladie, pour désigner même un seul cas ». Ils disent de cette définition qu'elle est la plus faible. Une autre définition donnée à l'épidémie par les auteurs est : « tout nombre identifié de cas dans un espace donné ou dans un temps donné ou alors en prenant en compte les deux. ». La troisième définition que nous allons relayer ici est « une augmentation de l'occurrence d'une maladie dans une population définie qui est clairement supérieure au nombre habituellement ou normalement observé dans cette population ».

Le point commun de ces trois définitions et de la conception populaire et intuitive que nous pouvons avoir de l'épidémie est « le nombre de cas ». Une épidémie est forcément liée au nombre de personnes souffrant de la maladie (Jacobson, et al., 1997). Plus important est ce nombre, plus grave est l'épidémie. Plus l'épidémie se propage rapidement plus ce nombre croit considérablement.

Voilà pourquoi dans la lutte contre une épidémie, la diminution de sa propagation est toujours le premier combat (Ainsworth & Over, 1997), (Birge, et al., 2020). Raison pour laquelle nous avons décidé de nous intéresser à cette problématique dans ce chapitre de thèse. En effet, l'objectif socio-sanitaire de ce travail, est de proposer un modèle enrichi de Machine Learning (ML) qui va contribuer à la lutte contre la propagation des épidémies. Pour ce faire, nous nous sommes fixés pour but que notre modèle soit moyennement plus performant que les modèles existant dans la littérature en ce qui concerne la détection des personnes atteintes d'une épidémie ; ainsi, il ne laissera pas s'échapper beaucoup de cas positifs qui peuvent aller contaminer d'autres et augmenter l'incidence de l'épidémie.

Pour arriver à atteindre cet objectif, nous avons d'abord parcouru la littérature pour découvrir les travaux des autres chercheurs IT dans la lutte contre les épidémies. Les résultats de nos découvertes sont présentés dans la section 6.2.1 de ce chapitre. Par la suite, nous avons choisi la maladie à corona virus 2019 (covid-19) comme l'épidémie sur laquelle nous testerons notre modèle. La description du dataset, les modèles ML utilisés pour comparer les performances du

modèle que nous proposons, ainsi que les critères d'évaluation (métriques) utilisés sont présentés dans la section 6. 2.2 de ce présent chapitre.

La section 3, quant à elle, présente les résultats obtenus pendant l'exécution de la méthodologie de recherche. Ces résultats comprennent, le prétraitement, l'analyse et le fractionnement des données, quelques algorithmes dont celui que notre modèle implémente, la construction de notre modèle, les résultats de l'exécution de notre modèle tenant compte de chaque étape de notre algorithme ainsi que les résultats de l'évaluation des métriques comparés aux autres modèles. C'est dans la section 4 que nous apportons une argumentation aux résultats présentés dans la section 3. Cette discussion permet de montrer comment notre modèle parvient à satisfaire l'objectif socio-médical de ce travail de recherche. Nous partons d'une analyse des valeurs des métriques obtenues par notre modèle en comparaison avec les autres modèles pour faire le lien avec la lutte contre la propagation de l'épidémie. La conclusion de ce chapitre est présentée dans la section 5.

6.2. Démarche

Ce chapitre de thèse fait suite au chapitre 4 qui a justifié le choix de Random Forest et au chapitre 5 qui a donné un formalisme de cette méthode. Ce chapitre a pour but de présenter la principale contribution de cette thèse qui consiste à proposer un modèle enrichi de Random Forest basé sur une version modifiée de l'algorithme de base de RF. Comme dit dans l'introduction, l'objectif de notre algorithme et de notre modèle est de réduire la propagation de l'épidémie.

Pour obtenir le résultat escompté, nous avons suivi une méthodologie de recherche en neuf étapes :

- **Choix de l'épidémie** : Il s'agit ici de déterminer quelle épidémie servira d'exemple d'expérimentation.
- **Choix du dataset** : Après avoir choisi l'épidémie, il faut par la suite choisir le dataset parmi ceux disponibles dans la littérature. C'est sur ce dataset que sera testé le modèle proposé.
- **Sélection des modèles ML** : Pour attester de la performance de notre modèle, il faudra impérativement le comparer aux autres. D'où l'importance de cette étape.
- **Modèle enrichi de RF** : Nous expliquerons dans cette étape le processus qui a conduit à la construction de notre modèle.

- **Prétraitement et Analyse de données** : C'est l'étape qui permet de préparer les données à être utilisées puis effectuer les études de corrélations pour mieux comprendre nos données avant la prédiction.
- **Fractionnement du dataset** : Consiste à diviser le dataset en deux parties. Une destinée à l'entrainement des modèles et l'autre à tester ces derniers.
- **Entrainement des modèles** : Il s'agit ici d'entrainer tous les modèles, y compris celui que nous proposons. Avant cela, les modèles à utiliser sont sélectionnés, et le modèle enrichi RF proposé.
- **Test modèles** : Il s'agit ici de tester tous les modèles, y compris celui que nous proposons.
- **Evaluation modèles** : Il s'agit ici d'évaluer les modèles sur la base des métriques définies.

Cette démarche est présentée dans la Figure 6.1.

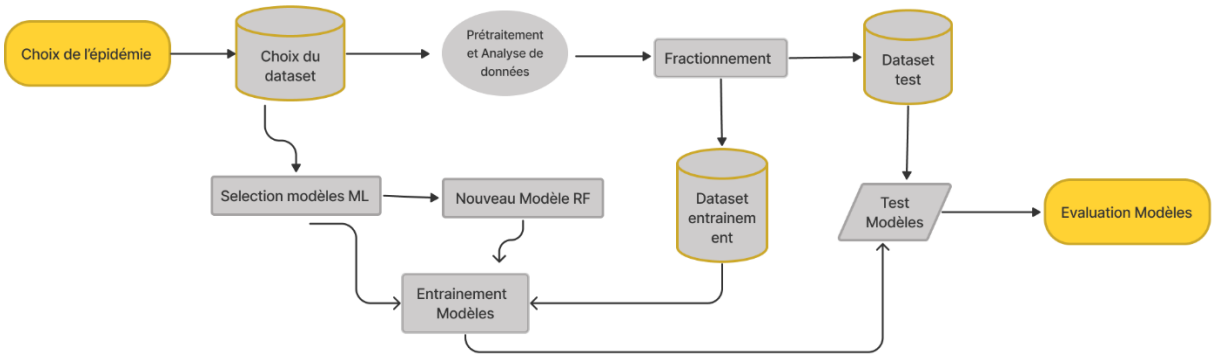


Figure 6.1 : Méthodologie de recherche

6.2.1. Travaux Connexes

La lutte contre l'épidémie de la covid-19 revêt plusieurs challenges selon différents aspects. Depuis les débuts de l'année 2020 les chercheurs du monde ont tenté de résoudre chaque aspect de cette lutte. Toutefois, l'objectif principal que les gouvernements et dirigeants des pays du monde se sont donnés dans la lutte contre la covid-19, est de limiter la propagation de la maladie. C'est aussi ce qu'affirme (Ahamad, et al., 2020).

(Ahamad, et al., 2020) ont implémenté et appliqué des modèles qui permettent de prédire et sélectionner les caractéristiques (variables) qui déterminent correctement si une personne est positive ou négative. Autrement dit, ces modèles permettent de déterminer l'ordre des

significativités des variables. Sur la base des variables et données sur lesquelles Ahamad M. et al. ont appliqué ces modèles, ils ont obtenu le résultat suivant : fièvre (41.1%), toux (30.3%), infection pulmonaire (13.1%) et écoulement nasal (8.43%). (Rustam , et al., 2020) ont effectué une étude afin de prédire le nombre de nouvelles personnes contaminées, mortes et rétablies dans les dix prochains jours. Les modèles développés dans (Rustam , et al., 2020) ont été construits sur la base des quatre méthodes suivantes : Linear Regression (LR), Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Machine (SVM), et exponential smoothing (ES). Les résultats obtenus par les auteurs ont prouvé que la méthode ES est la meilleure parmi les quatre, suivie de LR, LASSO puis SVM.

(Greco, et al., 2022) ont de leur côté mené une étude pour prédire le taux de mortalité des patients covid-19. Leur étude avait pour but de prédire la sortie de crise dans le service des soins intensifs. A l'issue de l'étude, ils ont trouvé que les caractéristiques suivantes étaient fortement liées à la mortalité : l'âge, le nombre des comorbidités et le genre masculin. (Muhammad, et al., 2020) se sont penchés sur la prédiction du rétablissement des patients de la covid-19. Ils ont développé des modèles avec quatre algorithmes. Le modèle construit avec DT s'est avéré meilleur que les autres avec une précision de 99, 85%. (Narin, et al., 2020) ont construit un modèle se basant sur CNN (Convolutional Neural Networks). Le modèle construit utilise les images x-ray pour prédire la contamination à la covid-19 à un stade préliminaire. Dans le même ordre d'idée, (Ozturk, et al., 2020) ont, dans leur travail de recherche, proposé un modèle qu'ils ont nommé DarkCovidNet qui permet de prédire la contamination à la covid-19 à partir des images de scanner thoracique.

Il y a eu aussi des études portant sur le niveau de propagation de l'épidémie, comme celle de (Amar, et al., 2020). Ils étudient dans leur travail l'expansion future de la covid-19, le moment probable où l'épidémie atteindra son pic ainsi que le moment où elle pourrait prendre fin. Se basant sur un dataset de l'Egypte, les auteurs ont prédit le résultat suivant : En Egypte, l'épidémie pourrait atteindre jusqu'à 166 760 cas avec un pic le 22 juin 2020 et une fin le 08 septembre 2020. (Yang, et al., 2020) ont construit un modèle permettant de prédire le pic et la taille de la pandémie en Chine. (Dianbo, et al., 2020) ont développé un modèle pour la prédiction en temps réel du nombre de personnes atteintes de la covid-19 en Chine. (Remuzzi & Remuzzi, 2020) ont effectué une étude pour prédire l'expansion de la pandémie en Italie et son impact en Chine. Les études comme celle de (Petropoulos & Makridakis, 2020) et celle de (Grasselli, et al., 2020) se sont intéressées à la prédiction en temps réel des cas de contamination

à la Covid-19 dans le monde ainsi que les premières réponses qui pourraient être apportées à ces cas.

D'autres travaux se sont focalisés sur des modèles supervisés de Machine Learning afin de prédire si une personne est positive ou négative à la covid-19. Citons dans ce registre, les travaux de (Muhammad, et al., 2021) et de (Buvana & Muthumayil, 2021). Dans (Muhammad, et al., 2020), les auteurs ont utilisé un dataset du Mexique comprenant les variables suivantes : âge, sexe, pneumonie, diabète, asthme, hypertension, maladie cardiovasculaire, obésité, maladie rénale chronique, tabac et résultat. Pour effectuer la prédiction, les auteurs ont utilisé et évalué les modèles suivants : Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM) et Artificial Neural Network (ANN). Ils sont arrivés à la conclusion suivante : DT s'est avéré être le meilleur modèle en ce qui concerne la métrique 'accuracy' avec une valeur de 94,99% suivi de LR (94,41%), NB (94,36%), SVM (92,40%) et ANN (89,20), alors qu'en ce qui concerne la métrique 'sensitivity', c'est SVM qui prend la première place avec 93,34% suivi de ANN (92.40%), DT (89, 20%), LR(86, 34%) et NB (83,76%), et enfin en ce qui concerne la métrique 'specificity', la première position est détenue par NB avec 94.30 % suivi de DT (93,22%), LR (87,34%), ANN (83,30%) et SVM (76,50%). Bien que dans cette étude les modèles se répartissent le leadership de performance selon les métriques, nous pouvons aisément constater qu'en moyenne le modèle DT prend la première place avec 92, 47% suivi de NB avec 90, 81%, LR avec 89,36%, ANN avec 88, 30% et enfin SVM avec 87,41%.

Dans (Buvana & Muthumayil, 2021) les auteurs ont utilisé un dataset publiquement accessible qui comporte les variables suivantes : pays, âge, genre, fièvre, douleur_corporelle, nez_coulant, difficulté_respiration, congestion_nasale, mal_gorge, sévérité et contact_avec_patient_covid. Pour leur étude, les auteurs ont utilisé les modèles suivants : GNB (Gaussian Naives Bayes), LR (Logistic Regression), SVM (Support Vector Machine), KNN (K-Nearest Neighbor), DT (Decision Tree). Les auteurs ont utilisé quatre métriques d'évaluation, 'Accuracy', 'Precision', Recall', 'F1 Score' et ont abouti au résultat selon lequel DT est le modèle le plus performant sur toutes les métriques.

Eu égard à ce qui précède, nous pouvons constater que les deux dernières études que nous avons cités à savoir (Muhammad, et al., 2021) et (Buvana & Muthumayil, 2021), placent DT en première position des meilleurs modèles pour la prédiction des épidémies, plus particulièrement en ce qui concerne la détermination du statut positif ou négatif à la Covid-19 des patients. Ces

études sont un fort point d'appui pour la démonstration empirique de l'étude (Bokonda, et al., 2021) que nous avions publié en janvier 2021. Cette étude était une brève revue de littérature qui avait pris en considération quatre épidémies : la Peste Porcine Africaine (PPA), la dengue, la grippe, et le norovirus de l'huître. A l'issue de cette étude Random Forest s'est distingué des autres comme étant le meilleur classifieur dans la prédiction des épidémies suivi de ANN (Artifical Neural Network).

Nous nous étions basés, à l'époque dans cette étude (Bokonda, et al., 2021), sur les travaux suivants : (Tapak, et al., 2019), (Liang, et al., 2020) ainsi que celui de (Ducharme, 2018). Chacun de ces trois travaux, dans des contextes différents, a abouti à la conclusion selon laquelle Random Forest était le meilleur classifieur. Nous avons Tapak L. et al. qui ont fait une comparaison de trois méthodes ML (SVM, ANN et RF) afin de juger leur capacité dans la prédiction des épidémies. Ils en viennent à la conclusion selon laquelle la prédiction temporelle du RF est meilleure par rapport à celles des autres méthodes étudiées (ANN et SVM) pour des problèmes de ce type alors qu'ANN est meilleure dans la détection des épidémies. Liang et ses co-auteurs testent et comparent six méthodes ML (Bayes Net, ANN, SVM, AdaBoost, C4.5 et RF) sur la base de quatre mesures de comparaison bien connues : SN (sensitivity), SP (specificity), ACC (accuracy) & MCC (Matthew Correlation Coefficient) avant d'opter pour le RF pour effectuer l'analyse prédictive de l'épidémie de la Peste Porcine en Afrique, parce que le résultat de cette comparaison a placé le RF au-dessus des autres. Ducharme compare six méthodes (Bn, RLog, SVM, RF, KNN, ANN). A l'issue de son étude, Ducharme parvient à la conclusion suivante : « Le classifieur qui ressort de cet exercice avec les meilleurs scores est RF et ses variantes. ».

Cette fois, le fait que DT se positionne en première position dans (Muhammad, et al., 2021) et (Buvana & Muthumayil, 2021), sachant que RF est une amélioration de DT car RF est un ensemble d'arbres de décision, est une preuve supplémentaire de la qualité de prédiction de RF dans le cas des épidémies.

A cet effet, notre étude a pour objectif d'évaluer la capacité prédictive de Random Forest sur la base des quatre métriques présentées dans la section suivante et de proposer par la suite un modèle enrichi de RF qui permet d'améliorer les dites métriques. Nous nous baserons ainsi sur l'étude de (Buvana & Muthumayil, 2021). La valeur ajoutée socio-médicale de cette étude est de proposer un modèle ML qui contribue à la diminution de la propagation de la maladie decorona virus.

6.2.2. Matériels et Méthodes ML

Dans cette section, nous présenterons et expliquerons les données, les méthodes Machine Learning (ML), les métriques et le workflow que nous avons suivi dans le cadre de notre étude ainsi que les autres outils.

6.2.2.1. Description du dataset

Les données utilisées viennent d'un dataset publiquement accessible sur Github (Simran, 2020). Ce dataset a déjà été utilisé précédemment par d'autres chercheurs dont Buvanaet Muthumayil dans (Buvana & Muthumayil, 2021). Cette étude qui a utilisé ce dataset a été publiée sur le site de l'Organisation Mondiale de la Santé (OMS) (WHO, 2021), ce qui a renforcé la crédibilité du dataset à nos yeux et nous a encouragé à l'utiliser. Ce dataset a été construit et mis en ligne par Simran Pandey (Simran, 2022), chercheuse et membre du centre de recherche IDC (Industrial Design Center) de l'Institut Indien de Technologie (IIT Bombay) (Industrial Design Centre, 2022).

Ce dataset contient de données démographiques et cliniques reparties en douze variables notamment : Country, Age, Gender, fever, Bodypain, Runny_nose, Difficulty_in_breathing, Nasal_congestion, Sore_throat, Severity, Contact_with_covid_patient et Infected ainsi que 2500 entrées.

6.2.2.2. Méthode ML

Ce travail de recherche vise à apporter une contribution à la résolution d'une problématique qui se classe volontiers dans la catégorie classification supervisée. Commençons donc par préciser que pour résoudre un problème, l'apprentissage supervisé est utilisé lorsque les données historiques entrées/sorties sont connues. Le système est au préalable entrainé avec ces données puis utilisé par la suite pour trouver les sorties pour de nouvelles entrées (Kaur & Kumari, 2018). Le problème est dit de classification lorsque la variable à prédire est une variable catégorielle. Notre problème remplit pleinement ces critères. Dans nos trente mille données, nous avons aussi celle de la variable « Infected » qui est la variable à prédire et il s'avère que c'est une variable catégorielle qui permet de déterminer si une personne est positive ou négative.

Sur cette base, nous nous sommes orientés vers les méthodes de Machine Learning (ML) de type supervisé pouvant faire de la classification. Les modèles considérés dans cette étude ont

été construit sur la base des six méthodes ML suivantes : GNB (Gaussian Naives Bayes), LR (Logistic Regression), SVM (Support Vector Machine), KNN (K-Nearest Neighbor), DT (Decision Tree) et RF (Random Forest).

1. Gaussian Naives Bayes

Gaussian Naives Bayes est une variante de Naives Bayes qui se base sur le théorème de Bayes. Le théorème de Bayes permet de trouver les probabilités conditionnelles d'occurrence de deux évènements sur la base des probabilités d'occurrence de chaque évènement selon la formule suivante :

$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right)P(A)}{P(B)}$$

Avec :

P(A/B) = La probabilité que l'évènement A soit vrai sachant que B est vrai.
P(B/A) = La probabilité que l'évènement B soit vrai sachant que A est vrai.
P(A) = La probabilité que A soit vrai.
P(B) = La probabilité que B soit vrai.

Lorsque Gaussian Naives Bayes est utilisé pour des cas de classification basés sur plusieurs variables comme c'est le cas pour nous, une hypothèse forte est posée : les variables sont considérées indépendantes les unes des autres (Ontivero-Ortega, et al., 2017).

2. Logistic Regression

Logistic Regression est une méthode statique souvent utilisée pour la classification et l'analyse prédictive (Asadi, et al., 2014), (Ayyoubzadeh, et al., 2020). Son algorithme se base sur des variables indépendantes afin de prédire la probabilité qu'un événement puisse arriver. La prédiction étant une probabilité, elle est comprise entre 0 et 1 (Edison , et al., 2020). Pour une classification binaire, lorsque le résultat est inférieur à 0,5 la prédiction sera 0 au cas contraire elle sera 1 (Ishaq, et al., 2018).

Pour arriver à donner le résultat escompté l'algorithme de logistic regression applique une transformation connue sous le nom de log odds. Cette transformation logistique se fait selon les formules mathématiques suivantes :

$$Logit(pi) = \frac{1}{(1 + e^{(-pi)})}$$

$$ln\left(\frac{pi}{1-pi}\right) = Beta_0 + Beta_1 * X_1 + \cdots + B_k * K_k$$

Avec :

Logit(pi) : La variable cible dite aussi variable dépendante.

X : La variable indépendante.

Beta : Un coefficient estimé grâce au maximum de vraisemblance (MLE).

3. Support Vector Machine

Support Vector Machine (SVM) est utilisé à la fois pour la classification et pour la régression mais aussi pour la détection d'anomalie (Noble, 2006). Dans notre cas, comme dans la plupart des cas, il est utilisé pour la classification. L'algorithme du SVM traite l'ensemble des données comme un ensemble de points et se donne pour objectif de trouver l'hyperplan optimal qui divise cet ensemble en deux groupes (deux classes) dans le cas d'une classification binaire, comme la nôtre. Tout le challenge consiste à trouver, parmi tant d'hyperplans possibles, celui qui divise le mieux les deux classes de tel sorte que la frontière des classes soit la plus éloignée possible des points de données, les Figures 6.2 et 6.3 illustrent ce processus. Il s'agira donc de maximiser la marge (Mammone, et al., 2009). La marge étant définie comme la distance entre le point de donnée le plus proche et l'hyperplan.

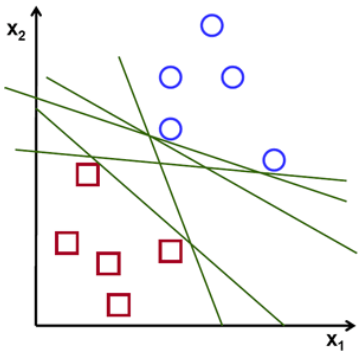


Figure 6.2 : SVM : hyperplans possibles

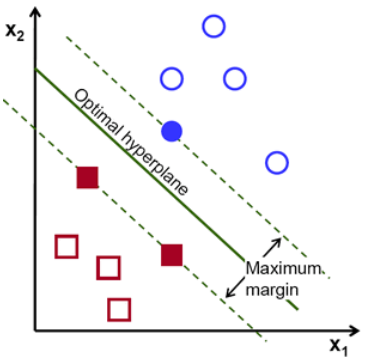


Figure 6.3 : SVM : hyperplan optimal

4. K-Nearest Neighbor

K-Nearest Neighbor (k-NN) est un classifieur supervisé non paramétrique (Kramer, 2013). Il peut être utilisé pour les cas de classification, mais aussi ceux de régression, bien que cette dernière utilisation soit peu fréquente. Pour la classification, k-NN fonctionne sur la base de l'hypothèse selon laquelle les points similaires peuvent se retrouver les uns à côté des autres (Bezdek, et al., 1986). Partant de cette hypothèse, une étiquette est attribuée à un point selon qu'elle est majoritaire autour de celui-ci.

Autrement dit, si un point x_1 est entouré par les points x_2, x_3, et x_4 supposons que x_2, x_3 appartiennent à la classe « A » et que x_4 appartienne à la classe « B » alors x_1 sera considéré comme de la classe « A ». La Figure 6.4 illustre bien ce fonctionnement.

Figure 6.4 : Fonctionnement de k-NN

5. Decision Tree

Décision Tree (DT) est un classifieur supervisé non-paramétrique pouvant être utilisé pour la classification et la régression. Les arbres de décision sont construits en suivant la stratégie « divide and conquer » à partir d'un ensemble de données (Quinlan, 1996). En effet, un arbre de décision est une structure hiérarchique organisée en trois principaux éléments : le nœud racine, les nœuds internes et les nœuds feuilles.

Au départ toutes les données sont placées dans le nœud racine qui sera par la suite divisé en deux ou plusieurs nœuds dits nœuds internes ou nœuds de décision sur la base d'une règle (Suthaharan, 2016). Les nœuds internes seront à leur tour divisés pour en former d'autres, ainsi de suite, de manière récursive, jusqu'à ce qu'il ne soit plus possible de les diviser. Les derniers nœuds sont appelés nœuds feuilles ; ce sont eux qui constituent les prédictions finales.

Pour chaque nœud S (interne ou feuille), son estimation (étiquette dans le cas d'une classification) par rapport à la variable cible est calculé sur la base de l'entropie (Quinlan, 1986), (Li, et al., 2009) :

$$H(S) = -\sum_{i=1}^{m} p_i log(p_i)$$

Avec :

H(S) : L'entropie du nœud S. Permet de déterminer l'homogénéité du nœud. Plus elle tend vers zero (0) plus le nœud est homogène.

p_i : La probabilité qu'un élément de S se retrouve dans la classe C_i.

6. Random Forest

Apparu dans les années 90 Random Forest est une méthode ensembliste fonctionnant selon deux principes de base : le bagging et le Random Feature Selection (Shlien, 1990) , (Ho, 1995), (Breiman, 2001), (Breiman, 1996). Leo Breiman, dans (Breiman, 2001), définit une forêt aléatoire comme un classifieur constitué d'un ensemble de classifieurs élémentaires de type arbres de décision, notés :

$$\{ h(x, \Theta_k), k = 1, ...L \}$$

Avec :

$\{ \Theta_k \}$: Une famille de vecteurs aléatoires indépendants et identiquement distribués, et au sein de laquelle chaque arbre participe au vote de la classe la plus populaire pour une donnée d'entrée x.

En effet, Random Forest bénéficie de la simplicité des arbres de décision tout en corrigeant leur grande faiblesse qui est le surapprentissage. Il s'agit là d'une amélioration des arbres de décision. Random Forest a été conçu pour être plus robuste et plus précis qu'un arbre de décision.

6.2.2.3. Critères d'évaluation

L'objectif de ce chapitre est d'évaluer la performance des modèles ML dans la prédiction des épidémies pour des cas de classification avant de proposer un modèle enrichi. Pour arriver à

cette fin, nous avons considéré principalement quatre métriques qui sont accuracy, precision, recall et f1 score, pour évaluer six modèles : GNB, LR, SVM, KNN, DT et RF. Par la suite pour comparer le modèle RF par défaut et le modèle RF proposé nous avons considéré, en plus des quatre métriques précédents, la matrice de confusion. Dans la suite, nous allons définir les métriques utilisées dans une perspective épidémiologique.

1. Accuracy

Accuracy est l'une des métriques les plus utilisées. Elle permet de mesurer la précision du modèle sur l'ensemble des prédictions. En effet, Accuracy est une mesure à la fois sur les prédictions positives et négatives. Elle permet de mesurer simultanément le nombre des vrais positifs et vrais négatifs (Grandini, et al., 2020). Sa formule est la suivante :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Avec :

TP (True Positive) : Les prédictions positives et qui sont réellement positives.
TN (True Negative) : Les prédictions négatives et qui sont réellement négatives.
FP (False Positive) : Les prédictions positives mais qui sont en réalité négative.
FN (False Negative) : Les prédictions négatives mais qui sont positives.

2. Precision

Precision permet de mesurer le nombre de personnes déclarées positives par le modèle et qui le sont réellement par rapport au nombre total de personnes déclarées positives par le modèle (Grandini, et al., 2020). Autrement dit, c'est la métrique qui nous permet d'estimer le niveau de confiance que nous pouvons accorder au modèle lorsque celui-ci déclare une personne positive. Sa formule est la suivante :

$$Precision = \frac{TP}{TP + FP}$$

3. Recall

Aussi appelée Sensitivity, Recall est l'une des métriques incontournables de la classification. Elle permet de mesurer le nombre de personnes déclarées positives par le modèle et qui le sont

réellement par rapport au nombre total de personnes positives dans le jeu de données (Grandini, et al., 2020), (Eusebi, 2013). Autrement dit, c'est la métrique qui permet de mesurer la capacité du modèle à détecter tous les cas positifs qui lui sont transmis. Il permet de répondre à la question suivante : sur tous les enregistrements positifs, combien ont été correctement prédits ? Un modèle avec un Recall élevé laissera échapper moins de cas positifs. Sa formule est la suivante :

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

F1-Score permet aussi de mesurer la performance des modèles ML. En effet, F1 Score est la moyenne pondérée de Precision et Recall (Grandini, et al., 2020). F1 Score permet de trouver le meilleur compromis entre la précision et le Recall (Sasaki, 2007). Sa formule est la suivante :

$$F1Score = \left(\frac{2}{precision^{-1} + recall^{-1}}\right) = 2\left(\frac{precision * recall}{precision + recall}\right)$$

5. Confusion Matrix

La matrice de confusion ou tableau de contingence, présenté sur la Figure 6.5, ne constitue pas en elle une métrique de performance. Elle est par contre un bon moyen pour visualiser toutes les métriques précédemment définies.

Elle permet, via un support visuel (tableau), d'observer à quelle fréquence les prédictions ont été bonnes par rapport à la réalité (Susmaga, 2004). La matrice de confusion permet de visualiser directement sur un tableau par exemple le nombre de personnes correctement prédites positives par rapport au nombre total des prédictions dans le jeu de données. Il ne s'agira plus uniquement de mesurer une métrique (accuracy, recall, etc.) ou le taux d'erreur, mais plutôt d'avoir des chiffres précis liés au cas traité (Susmaga, 2004).

Nous l'avons utilisée dans notre cas pour visualiser le résultat entre le modèle RF par défaut et le modèle RF que nous proposons.

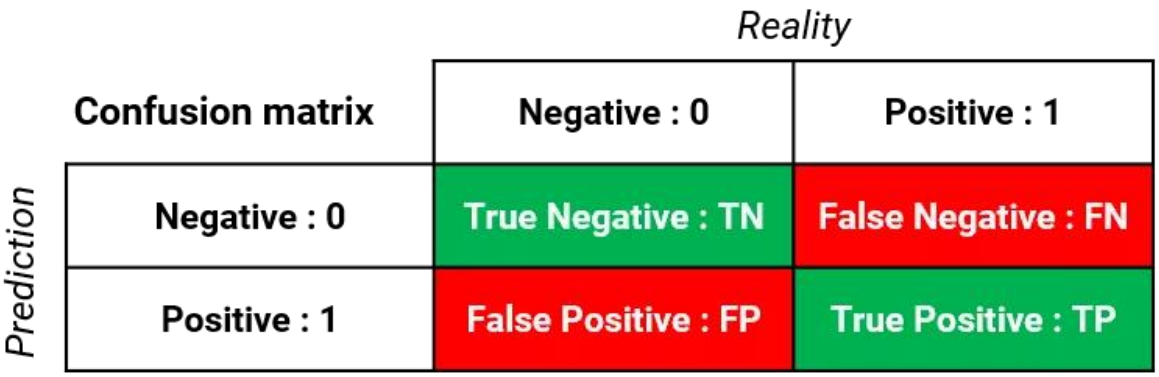


Figure 6.5 : Matrice de confusion

6.3. Résultats

Cette section a pour vocation de présenter le résultat obtenu après application de la méthodologie de recherche résumée par la Figure 6.5. Rappelons que dans cette étude, nous avons pour objectif principal de mettre en place un algorithme de classification à base des forêts aléatoires (Random Forest), pour permettre le test de Covid-19 en se basant sur certaines informations fournies par le patient à l'IA (Intelligence Artificielle) qui s'en servira pour prédire le résultat. Premièrement, nous commencerons par les statistiques descriptives sur les variables quantitatives, nous mènerons une analyse de corrélation entre la variable d'intérêt (variable cible) et les autres variables. Ensuite, nous expliquerons les algorithmes de bases de Random Forest par défaut et celui que nous avons modifié. Enfin nous élaborerons une comparaison entre les modèles utilisés par l'article de référence (Buvana & Muthumayil, 2021), RF par défaut, et notre modèle qui implémente notre algorithme.

6.3.1. Epidémie et dataset

Par la force des choses, pour être en phase avec notre époque et contribuer aussi à la lutte contre la Covid-19, nous avons choisi de prendre la pandémie Covid-19 comme exemple sur lequel implémenter notre modèle.

Pour le choix du dataset, nous avons été conduits par les contraintes suivantes :

1. Un dataset construit pour une étude de classification.
2. Toutes les variables du dataset doivent être collectables via un formulaire (notamment le formulaire ODK (Loola Bokonda, et al., 2019)). Cette contrainte vient du fait que dans la suite de cette thèse, les données seront collectées via ODK puis rajoutées à ce

dataset pour pouvoir tester l'ajout du module de prédiction à ODK afin de former le MEPS (Mobile Epidemiological Prediction). Cette partie est traité dans le chapitre 7.

3. Le dataset doit avoir été utilisé dans un article scientifique.
4. L'article scientifique qui utilise ce dataset doit comporter une comparaison de plusieurs modèles Machine Learning (ML) sur la base des métriques citées dans la section 2.2.
5. Au meilleur des cas, parmi les modèles comparés dans l'article qui utilise ce dataset, il doit y avoir Random Forest.

L'issue de nos recherches nous a conduit vers le dataset décrit dans la section 6.2.2 et dont l'entête est représenté par la Figure 6.6. Toutes les variables de ce dataset peuvent être collectées via un formulaire, il a été utilisé par Buvana, M. et Muthumayil K. dans (Buvana & Muthumayil, 2021). Des cinq contraintes précédentes, ce dataset rempli pleinement les quatre premiers. S'agissant du cinquième critère (comparaison de RF et les autres modèles), la présence du modèle DT (Decision Tree) parmi les modèles comparés dans l'article, suffit pour combler le vide de RF parce qu'il existe un lien évident entre RF et DT.

Country	Age	Gender	fever	Bodypain	Runny_nose	Difficulty_in_breathing	Nasal_congestion	Sore_throat	Severity	Contact_with_covid_patient	Infected
China	10	Male	102	1	0	0	0	1	Mild	No	0
Italy	20	Male	103	1	1	0	0	0	Moderate	Not known	1
Iran	55	Transgender	99	0	0	0	1	1	Severe	No	0
Republic of Korean	37	Female	100	0	1	1	0	0	Mild	Yes	1
France	45	Male	101	1	1	1	1	0	Moderate	Yes	1

Figure 6.6 : Entête du dataset utilisée dans le cadre de cette étude

6.3.2. Prétraitement, analyse et fractionnement des données

L'analyse de données de notre dataset révèle que nous avons un échantillon de personnes dont l'âge varie entre 10 et 89 ans, comme le montre la Figure 6.7. La moyenne se situe à 43 ans et la médiane à 39 ans. Cependant, la majorité des personnes appartiennent à la tranche d'âge 35-55 ans.

En ce qui concerne la température corporelle des patients, elle est exprimée en Fahrenheit. Elle s'étale sur 98 à 204° F, comme le montre la Figure 6.8. La moyenne et la médiane sont d'environ 100°F avec un écart-type de 1.71. Nous observons que la majorité de nos sujets (entre la première quartile et la troisième) ont une température variant entre 99 à 102°F soit entre 37,22 à 38,88°C. Notons que Celsius= (Fahrenheit-32)/1.8.

La corrélation de la variable d'intérêt *Infected,* variable qui détermine si une personne est positive à la covid ou pas, avec les variables indépendantes révèle des informations intéressantes.

Dans le Tableau 6.1 qui résume ces corrélations, il paraît que le mode de transmission de covid est principalement par contact avec une personne positive avec une corrélation de 57%, et en même temps,

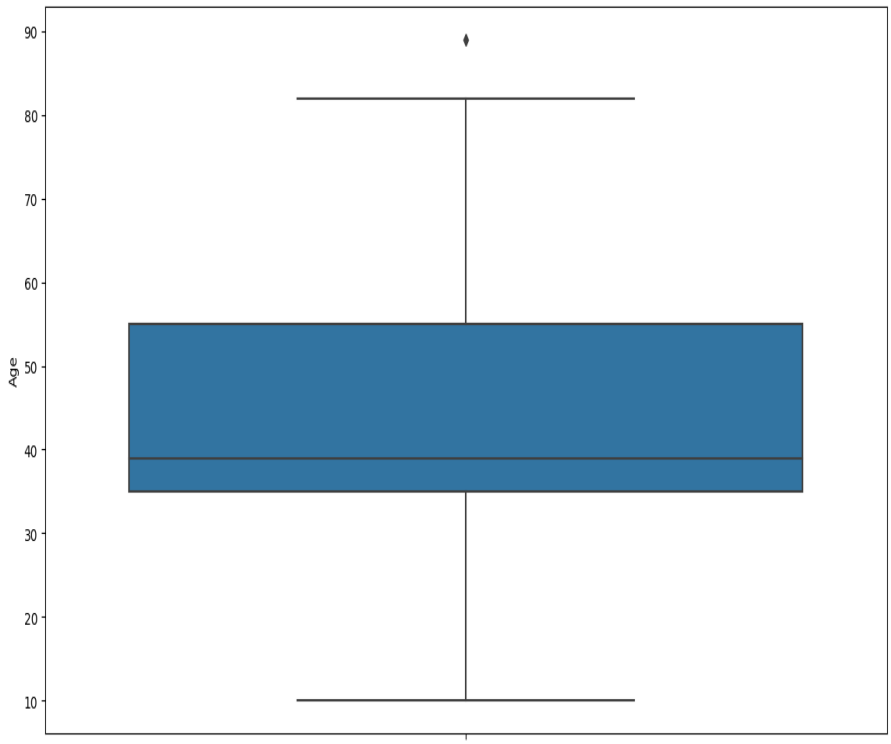


Figure 6.7 : Statistiques descriptives de la variable « Age »

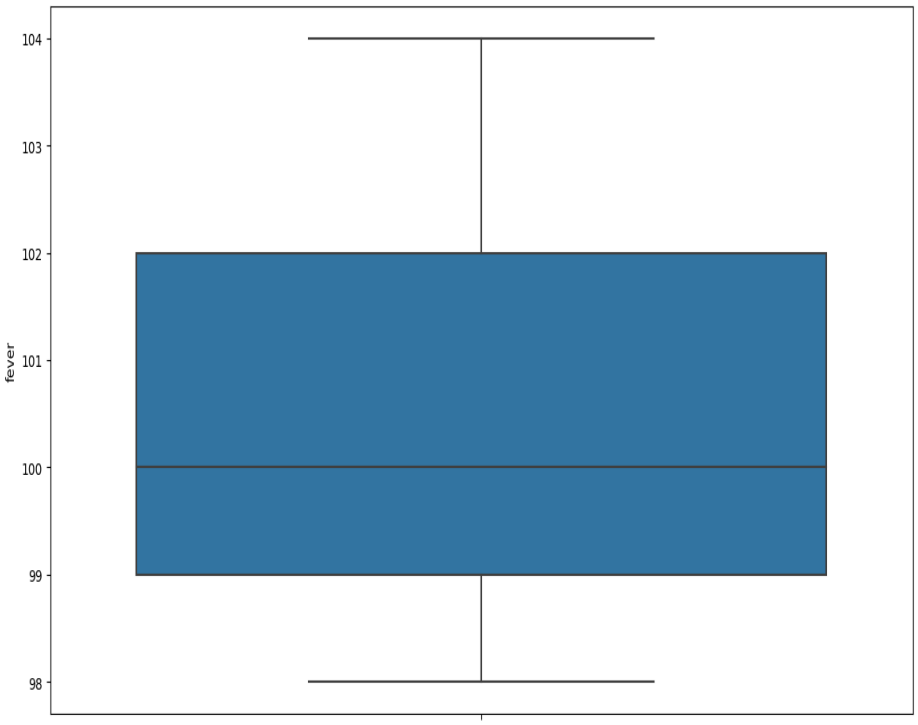


Figure 6.8 : Statistiques descriptives de la variable "Fever"

Il s'avère que le patient qui n'a pas eu de contact avec une personne positive a 80% de chance d'être sain. Les deux autres caractéristiques très influentes sont la difficulté de respiration et la courbature qui en cas de présence avérée conduisent généralement à un test positif à la covid à 48.3% et 44.2% respectivement. La fièvre anormale aussi est un indicateur conduisant à plus 39% des cas de présence du virus ; ce qui justifierait la prise de température dans les endroits très fréquentés comme la banque ou le centre commercial, car c'est l'un des moyens faciles de détecter des cas suspects de Covid-19.

Tableau 6.1 : Corrélations entre la variable cible et les variables indépendantes

Variables Indépendantes	Infected
Age	0.165
Fever	0.390
Bodypain	0.442

Runny_nose	0.284
Difficult_breathing	0.483
Nasal_congestion	0.287
Sore_throat	-0.218
Gender_Female	0.094
Gender_Male	-0.066
Severity_Mild	-0.368
Severity_Moderate	0.206
Severity_Severe	0.257
Contact_with_covid_patient_no	-0.796
Contact_with_covid_patient_not known	0.327
Contact_with_covid_patient_yes	0.579

Observons maintenant la distribution des données selon les deux classes : positif (personnes contaminées), négatif (personnes saines) respectivement 1 et 0. La Figure 6.9, montre que la distribution est équilibrée par classe permettant ainsi d'éviter le biais dû à la surreprésentation

d'une classe par rapport à l'autre provoquant un faible taux de précision du modèle pour la classe sous représentée.

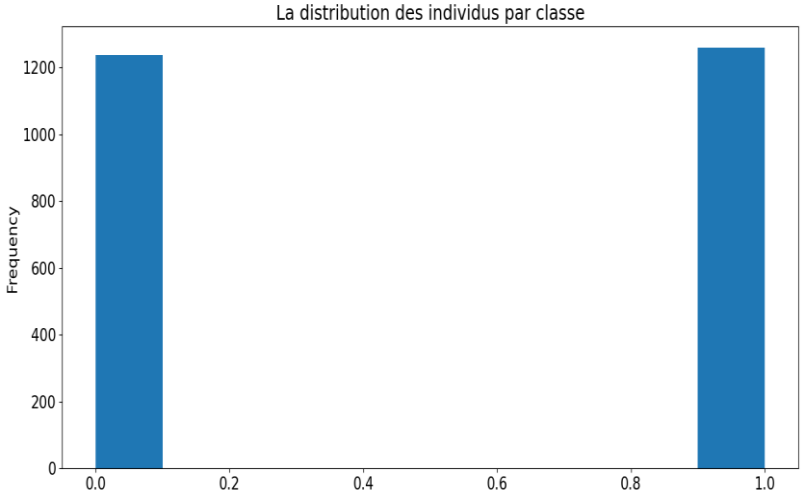


Figure 6.9 : Distribution des individus par Classe

Avant de passer à la mise en place de notre modèle, il nous a fallu adapter nos données à la forme compréhensible par notre modèle de Machine Learning c'est à dire l'encodage. Ce dataset était bien formaté et nettoyé lors de son téléchargement. La majorité des variables qualitatives étaient déjà encodées. Ainsi après l'exploration des données, nous avons procédé au codage des variables qualitatives qui n'étaient pas encore encodées. Pour éviter les effets d'échelles pour les variables ayant plus de deux valeurs nous avons opté pour l'encodage **one hot** aussi appelé *dummy encodage*, permettant de créer une nouvelle variable indicatrice pour chaque modalité. Ce fut le cas des variables « *Severity* » *et* « *Contact_with_covid_patient* ». L'encodage a eu lieu de la manière suivante :

Gender : Avec un encodage binaire 0/1 pour Male/Female.

Severity : L'encodage one hot de cette variable est illustré par la Figure 6.10. Il a permis d'exploser la variable en trois: *Severity_Moderate, Severity_Mild et Severity_Severe*. Les trois nouvelles variables ont subi à leur tour un codage binaire chacune : 0/1 pour absence/présence.

Contact_with_covid_patient : Le dummy encodage de cette variable a permis d'avoir les trois variables suivantes : *Contact_with_covid_patient_yes, contact_with_covid_patient_no et contact_with_covid_patient_ignore*. Les trois nouvelles variables ont subi à leur tour un codage binaire chacune : 0/1 pour absence/présence.

Les autres variable qualitatives (ou catégorielles) qui étaient déjà encodées sont : *Bodypain, Runny_nose, Difficulty_in_breathing, Nosal_congestion, sorethroat et Infected.*

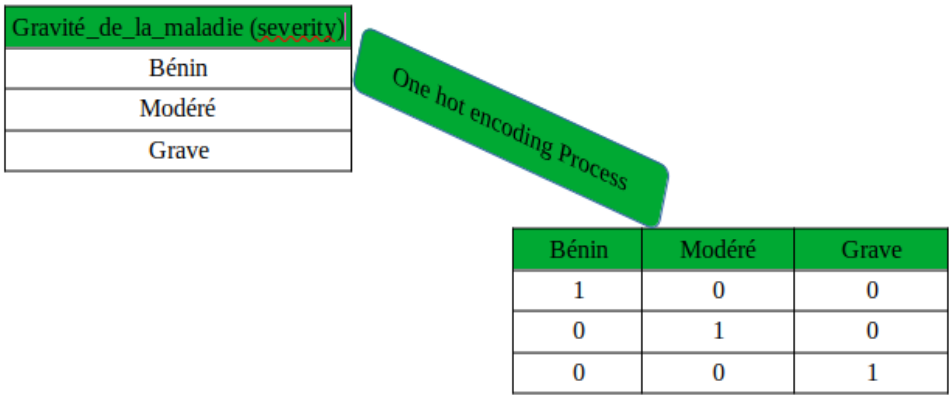


Figure 6.10 : Dummy encodage de la variable "Severity"

Pour entraîner le modèle, nous avons opté pour la séparation des données en adoptant 80 % pour entraîner le modèle et 20 % pour tester le modèle construit.

6.3.3. Entrainement, test et évaluation des modèles

Comme nous l'avons dit ci-haut, notre étude utilise un dataset qui a été utilisé par Buvana et Muthumayil dans (Buvana & Muthumayil, 2021). Nous allons nous baser sur les résultats qu'ils ont obtenu pour les cinq premiers modèles à savoir GNB, LR, SVM, KNN et DT puis comparer ces résultats à ce que nous avons obtenu avec Random Forest par défaut sur le même dataset puis avec Random Forest EP (Epidemiological Prediction) toujours sur le même dataset. Le Tableau 6.2 donne les résultats des métriques de six modèles dont cinq venant du travail de (Buvana & Muthumayil, 2021) et le sixième venant de notre travail.

Tableau 6.2 : Comparaison des métriques entre RF et autres modèles

Model	Accuracy	Precision	Recall	F1 score
GNB	0.799	0.786	0.828	0.806
LR	0.809	0.798	0.831	0.814
SVM	0.850	0.811	0.916	0.860
KNN	0.923	0.937	0.909	0.923
DT	0.945	0.939	0.952	0.946
RF par défaut	0.949	0.940	0.966	0.952

Nous appelons ici RF par défaut, la version RF présente par défaut dans le package scikit-learn. RF par défaut est construit sur la base de l'algorithme Forest RI proposé par Breiman L (Breiman, 2001) qui fait appel à l'algorithme récursif Random Tree. Forest RI est considéré

comme l'algorithme de référence de Random Forests (Bernard, et al., 2007). Afin de proposer un modèle enrichi RF, que nous avons choisi de nommer RF EP (Epidemiological Prediction), nous avons entrepris des modifications au cœur même de l'algorithme de base de Random Forest. Ces modifications ont conduit à un nouvel algorithme que nous avons nommé Forest EP qui fait appel à l'algorithme *Var Cust* que nous proposons en plus de *Random Tree* utilisé dans *Forest RI*. C'est sur la base de cet algorithme que nous avons construit *Random Forest EP*. Ces algorithmes et leurs explications sont donnés dans la suite de cette section.

Algorithme 1 : Forest RI

Entrée : T l'ensemble d'apprentissage

Entrée : L le nombre d'arbres dans la forêt

Entrée : K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud

Sortie : foret l'ensemble des arbres qui composent la forêt construite

1 : pour l de 1 à L faire

2 : Tl ← ensemble bootstrap, dont les données sont tirées aléatoirement (avec remise) de T

3 : arbre ← un arbre vide, i.e. composé de sa racine uniquement

4: arbre.racine ← RndTree(arbre.racine,Tl,K)

5 : foret ← foret ∪ arbre

6 : retour foret ;

Explication de Forest RI : Permet de rassembler l'ensemble des composantes de la forêt aléatoire. Avec L le nombre d'estimateur(arbre) à utiliser dans la construction de la forêt, pour chaque arbre nous faisons appel à la méthode de construction d'un arbre RnTree afin de construire l'arbre correspondant, en utilisant :

- Des données tirées aléatoirement à partir d'une base des données avec remise T_l
- Des variables ou features sélectionnées aléatoirement K

A chaque itération, un arbre est généré et les aléas utilisés pour les variables et dans les données avec Bootstrap permet de supposer l'indépendance d'opinion de ces estimateurs. Enfin ces arbres sont regroupés pour constituer la forêt dite aléatoire du fait de la méthode de construction des arbres.

Algorithme 2 : Random Tree

Fonction RndTree(n, T, K)
Entrée : n, le noeud courant
Entrée : T, l'ensemble des données associées au noeud n
Entrée : K, le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud
Sortie : n, le même noeud, modifié par la procédure
1. Si n n'est pas une feuille alors :
a. C ← K caractéristiques choisies aléatoirement
b. Pour chaque A appartenant à C **faire :**
i. Appliquer la procédure CART pour créer et évaluer (critère de Gini) le partitionnement produit par A, en fonction de T
ii. partition ← partition qui optimise le critère Gini
iii. n.ajouterFils(partition) #ajoute un nouveau nœud à l'arbre de décision en tant que fils du nœud courant n.
iv. Pour chaque fils appartenant à n.noeudFils **faire :**
1. fils ← RndTree(fils, fils.donnes, K)
2. Retourner n

Explication de Random Tree : Elle permet la mise en place d'un arbre à travers la méthode de partitionnement sur la base de variables aléatoirement choisies K. A partir d'un nœud n pour construire le nœud fils ou potentiellement une feuille nous appliquons pour chaque variable A ∈ C le critère de Gini afin de prendre la partition qui minimise le désordre ou qui permet de bien prédire la valeur réelle. Ce processus est ensuite utilisé sur chaque nœud pour construire l'arbre de manière récursive.

Algorithme 3 : Forest EP

Entrée : X_train l'ensemble d'apprentissage
Entrée : X_test l'ensemble de test
Entrée : y_train la variable d'intérêt d'apprentissage
Entrée : k le nombre de variable à éliminer de l'ensemble d'apprentissage
Entrée : threshold_ratio la quantité d'information à retenir dans les données d'apprentissage
Entrée : L le nombre d'arbres dans la forêt
Entrée : K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud
Sortie : foret l'ensemble des arbres qui composent la forêt construite

T_train_retain, T_test_retain= *Var_Cust(X_train, X_test, y_train, n, threshold_ratio)*

1 : pour l de 1 à L faire

2 : Tl ← ensemble bootstrap, dont les données sont tirées aléatoirement (avec remise) de T_train_retain

3 : arbre ← un arbre vide, i.e. composé de sa racine uniquement

4: arbre.racine ← RndTree(arbre.racine,Tl,K)

5 : foret ← foret ∪ arbre
6 : retour foret ;

Algorithme 4 : Var Cust

Entrée : n le nombre de variable moins significative à enlever de la base des données
Entrée : threshold_ratio la quantité d'information à garder dans les composantes principales
Entrée : X_train donnée d'entrainement
Entrée : X_test donnée de test
Entrée : y_train label des données d'entrainement
Sortie : X_train_retain les données d'entrainement après le traitement
Sortie :X_test_retain les données de test après le traitement

1. selector_fitted ← le nombre de variables sélectionnées par ordre de significativité.
2. X_scaled_train ← les données d'entrainement normalisées.
3. X_scaled_test ← les données de test normalisées.
4. X_pca_train ← réduction de dimension des données d'entrainement
5. X_pca_test ← réduction de dimension des données de test
6. X_train_retain ← X_pca_train. threshold_ratio
7. X_test_retain ← X_pca_test. threshold_ratio
8. **retour X_train_retain, X_test_retain**

Explication et résultat de l'algorithme Var_Cust lors de l'exécution du modèle RF EP :

[1] Cette étape extrait des variables initiales dans la base de données qui expliquent le mieux la variable d'intérêt (Infected), en respectant le nombre *n* donné en entrée.

```
Entrée [18]: X_train,X_test=randomclassi.var_cust(X_train_s,X_test_s,y_train,n=2,threshold_ratio=96)

             Les variables les plus significatives retenues après la selection des variables sont:  ['Age', 'Bodyp
             ain', 'Runny_nose', 'Difficulty_in_breathing', 'Nasal_congestion', 'Sore_throat', 'Severity_Mild', 'S
             everity_Moderate', 'Severity_Severe', 'Contact_with_covid_patient_no', 'Contact_with_covid_patient_no
             t known', 'Contact_with_covid_patient_yes']
```

Figure 6.11 : Exécution de l'algorithme var_cust étape 1

[2,3] Afin de poursuivre avec le traitement de nos données d'entrainement et de test, nous normalisons les données en mettant toutes variables quantitatives sur une même échelle pour éviter l'effet de taille qui biaise l'inertie (information) expliquée par les autres variables avec une échelle réduite. Cette normalisation est effectuée suivant la méthode centré réduite par **standardscaler.**

$$X_{normalisé}=(X-\mu)/\sigma \text{ (formule de normalisation)}$$

Avec X une variable quantitative, μ la moyenne des X et σ l'écart type

Covid-19 RandomForest V4 Dernière Sauvegarde : il y a 19 minutes (auto-sauvegardé)

View Insert Cell Kernel Widgets Help

▶ Exécuter Code

```
Les variables après réduction de dimension :
            F1        F2        F3        F4        F5        F6        F7  \
1412 -2.584893 -0.435703 -0.178095  0.085029  0.342967  0.042264 -0.053492
11   -2.319069 -0.188635  0.986978 -0.085248  1.126949  0.701966 -0.049381
2283  0.979795 -1.568683  0.032878 -2.223449 -2.020346 -1.330903  0.001856
1491  0.872639  0.961870 -1.975350  0.316060  1.131070  0.070891  1.779683
1587  0.440182 -0.150245 -0.720833 -1.425287 -0.666594  1.226607 -1.516084

            F8        F9       F10           F11           F12
1412 -0.053800  0.940575 -0.505906  2.383234e-15  3.417529e-15
11   -0.215846 -0.309562 -0.568723  3.575896e-17  1.391869e-15
2283  0.205804  0.434813  0.049357  1.763569e-15 -2.599633e-16
1491 -1.621807  0.879021  0.456600  1.705722e-15  1.637688e-15
1587  0.011413 -1.562493 -0.256200  3.339959e-15 -2.208178e-15
```

Figure 6.12 : Exécution de l'algorithme var_cust étape 4 et 5

[4,5] Après avoir normalisé les données, nous procédons à la réduction de dimension en créant des nouveaux composants à partir des variables retenues dans l'étape [1]**.** Chaque composant étant associé à une valeur permettant d'expliquer son apport.

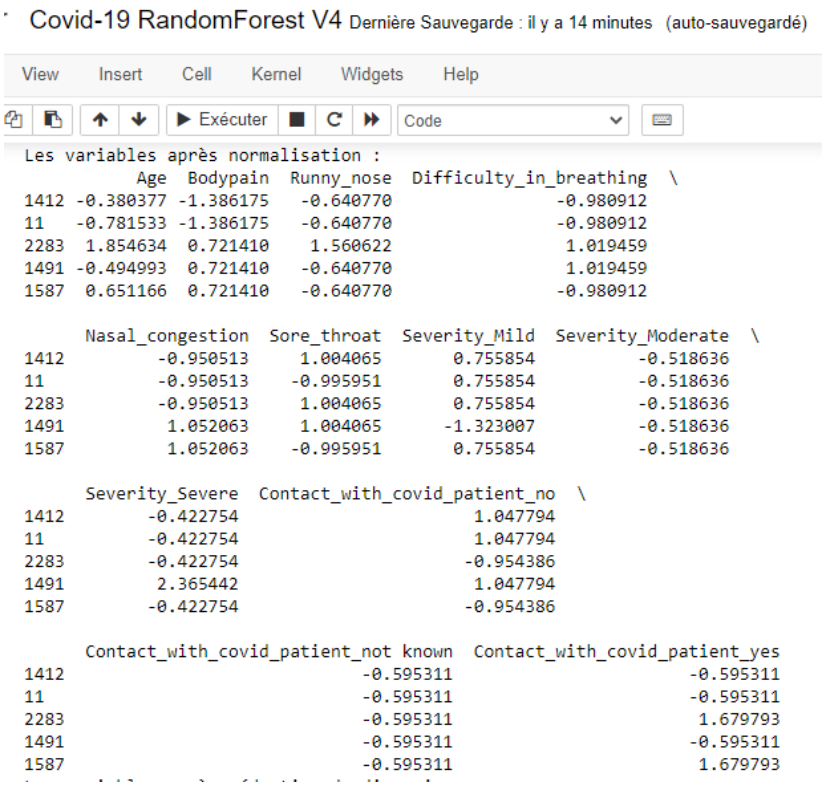


Figure 6.13 : Exécution de l'algorithme var_cust étape 2 et 3

[6,7] Enfin nous sélectionnons les nouvelles variables synthétisant le plus l'ensemble des informations de départ avec le choix de la quantité à retenir fixée par **threshold_ratio.**
L'implémentation de RF EP a donné des résultats satisfaisants présentés dans le Tableau 6.3.

```
Les variables retenues après selection des composants par rapport au ratio sont : ['F1', 'F2', 'F3',
'F4', 'F5', 'F6', 'F7', 'F8', 'F9']
```

Figure 6.14 : Exécution de l'algorithme var_cust étape 6 et 7

Tableau 6.3 : Comparaison des métriques entre RF EP, RF par défaut et les autres modèles

Model	Accuracy	Précision	Recall	F1 score
GNB	0.799	0.786	0.828	0.806
LR	0.809	0.798	0.831	0.814
SVM	0.850	0.811	0.916	0.860
KNN	0.923	0.937	0.909	0.923
DT	0.945	0.939	0.952	0.946
RF par défaut	0.949	0.940	0.966	0.952
RF EP	0.949	0.931	0.977	0.953

6.4. Discussion

L'objectif de cette étude est de contribuer à la diminution de la propagation de la maladie à corona virus. Pour y arriver, nous avons entrepris de proposer un modèle enrichi de Random Forest, nommé Random Forest EP (RF EP) avec des résultats qui nous permettront d'atteindre cet objectif. Notre modèle implémente le nouvel algorithme de Random Forest que nous avons proposé et nommé *Forest EP* qui est une modification de l'algorithme Forest RI, l'algorithme de base de Random Forest par défaut.

Afin d'évaluer l'efficacité de RF EP, nous nous sommes basés sur le travail de (Buvana & Muthumayil, 2021) puis sur une implémentation du RF par défaut que nous avons faite nous-même. Les résultats présentés dans le Tableau 6.2 dans la section 6.3.3, montrent que parmi les modèles évalués par (Buvana & Muthumayil, 2021), c'est DT qui se démarque comme le plus performant pour toutes les métriques considérées. Sur le même tableau nous avons inséré à la dernière ligne les résultats obtenus avec RF par défaut et nous pouvons observer, sans surprise, que RF est meilleur que DT pour toutes les métriques.

Les résultats qu'il est intéressant de commenter ici sont ceux de RF EP présentés dans le Tableau 6.3 de la section 3.3. La moyenne de toutes les métriques de RF par défaut est de 0.95175 et celle de RF EP est de 0,95250 soit environ une amélioration de 0,001. Bien plus que cette moyenne brute des métriques, il est plus intéressant d'analyser en détail l'évolution de chaque métrique.

Nous remarquons que les valeurs de la métrique *Accuracy*, sont pareilles dans les deux modèles (RF par défaut, et RF EP). Ce qui signifie que le nombre total de bonnes prédictions (positives et négatives) est la même dans les deux modèles. Cela peut se vérifier sur les matrices de confusion de RF EP et RF par défaut présentés par les Figures 6.15 et 6.16 respectivement. Au niveau de RF EP nous avons 213 TN + 255 TP = 468 et au niveau de RF par défaut nous avons 216 TN + 252 TP = 468.

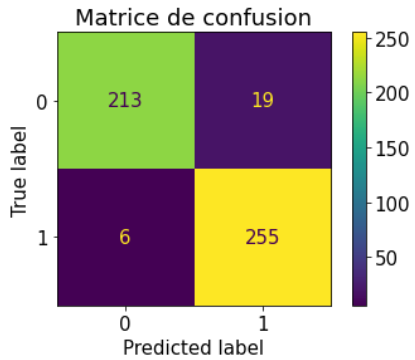


Figure 6.15 : Matrice de confusion de RF EP

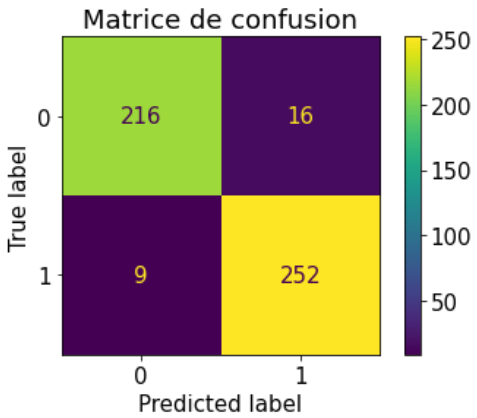


Figure 6.16 : Matrice de confusion de RF par défaut

La différence pourra donc être faite au niveau de la capacité à prédire correctement les cas positifs ou les cas négatifs. Pour cela, il faudra analyser les autres métriques.

Concernant la métrique *Precision*, RF EP enregistre une perte de performance de 0.009 par rapport à RF par défaut. Cela signifie que le test positif fourni par RF par défaut est plus fiable que celui fourni par RF EP. Comme les tests positifs sont plus fiables au niveau de RF par défaut, cela signifie que les tests négatifs le sont aussi. Par conséquent, RF par défaut donnera dans l'ensemble moins de faux tests négatifs (FN) par rapport à RF EP. Nous pouvons le vérifier sur les matrices de confusion de notre implémentation des deux modèles sur le dataset de test (test set) : RF par défaut a 16 FN alors que RF EP a 19 FN.

Le résultat de la métrique *Recall* montre un gain de performance de **0,011** au niveau de RF EP par rapport à RF par défaut. Ceci signifie que RF EP a une meilleure capacité prédictive concernant les cas positifs. En d'autres termes, lorsque nous considérons l'ensemble des cas positifs présents dans un dataset, avec RF EP nous allons détecter un plus grand nombre de cas par rapport à RF par défaut. RF EP permet ainsi de moins laisser s'échapper les patients atteints de la Covid-19. Or, ce sont les patients positifs qui véhiculent la maladie et propagent le virus, en augmentant la capacité du modèle à les détecter nous avons ainsi atteint l'objectif de cette étude. Ceci peut se vérifier sur les matrices de confusion de notre implémentation des deux modèles sur le dataset de test : RF par défaut a produit 252 TP et 9 FN alors que RF EP a produit 255 TP et 6 FN. Nous voyons ici que par rapport aux patients qui sont réellement positifs RF par défaut a mal classé trois personnes de plus que RF EP. Cela signifie, pour notre dataset, que trois porteurs de virus sont dans la nature en train de propager le virus alors qu'ils auraient pu

être détectés en utilisant RF EP. Le Score F1 qui est la moyenne harmonique des deux dernières métriques montre un gain de performance de **0,001** de RF EP par rapport à RF par défaut.

6.5. Conclusion

Ce chapitre avait pour objectif de proposer un modèle pour améliorer la lutte contre la propagation des épidémies. L'épidémie qui a été choisie comme exemple est la Covid-19. Pour ce faire, nous avons utilisé un dataset publiquement accessible sur Github et qui a été conçu pour une étude de type classification. L'analyse prédictive effectuée avait pour but de déterminer si une personne est positive ou négative à la Covid-19 sur la base de onze variables, notamment : *Country*, *Age, Fever, Bodypain, Runny_nose, Difficult_in_breathing, Nasal_congestion, Sore_throat, Gender, Severity et Contact_with_covid_patient*. La variable cible se nomme *Infected.*

La plupart des variables qualitatives de ce dataset était déjà encodées lorsque nous l'avons téléchargé à l'exception de *Gender, Severity et Contact_with_covid_patient* que nous avons encodé nous-même en utilisant un codage binaire pour *Gender* et le *dummy encodage* pour *Severity et Contact_with_covid_patient.*

Dans cette étude, nous proposons un modèle enrichi de RF. Le choix de RF s'est fondé sur une revue de littérature des travaux de recherche de plusieurs chercheurs qui ont comparé RF à d'autres modèles dans le contexte des prédictions des épidémies et ont placé RF en tête comme meilleur classifieur.

Le modèle enrichi de RF, nommé RF EP que nous proposons implémente un algorithme que nous avons proposé. Cet algorithme, nommé Var Cust, que nous avons placé au cœur même du code de RF par défaut qui devient ainsi RF EP permet d'effectuer quatre étapes au sein même de RF sans avoir besoin de chercher d'autres méthodes à l'extérieur. Ces quatre étapes sont : la sélection des variables significatives (un nombre n de variables moins significatif sera supprimé), la normalisation des données, la réduction de dimension, et enfin la sélection des nouvelles variables qui synthétisent le mieux l'information dont l'algorithme a besoin sur la base du **threshold_ratio** préalablement défini.

Somme toute, l'exécution de notre modèle a produit des résultats satisfaisants. Nous avons d'abord effectué une première comparaison entre RF par défaut et cinq autres modèles notamment : GNB, LR, SVM, KNN et DT. RF par défaut s'est démarqué de tous les autres modèles pour toutes les métriques considérées sachant qu'il y en avait quatre : Accuracy,

Precision, Recall, F1 score. En comparaison au RF par défaut, RF EP a une amélioration de performance de **0,011** sur la métrique Recall. Cette amélioration permet d'atteindre l'objectif fixé dès le départ de cette étude. En effet, ce gain de performance signifie que RF EP est en mesure de détecter plus de patients positifs que RF par défaut pour un dataset donné. Autrement dit, lorsqu'un hôpital ou un centre de santé utilise RF EP, ils augmentent leur capacité à détecter tous les patients positifs. En d'autres termes, moins de personnes porteuses de la maladie et donc susceptibles de la transmettre aux autres pourront passer par les mailles des tests. Ce résultat est appuyé par le test réel effectué sur notre dataset. Nous avons avec RF EP 255 TP et 6 FP alors avec RF par défaut nous avons 252 TP et 9 FP sur le même dataset.

Chapitre 7 : Système Mobile de Prédiction Epidémiologique

7.1. Introduction

Nous avons déjà, dans les précédents chapitres de cette thèse, discuté de tous les éléments nécessaires à la construction de ce que nous avons choisi de nommer dans ce chapitre le MEPS. MEPS pour Mobile Epidemiological Prediction System désigne une suite logicielle constituée des applications mobiles et conçue spécialement pour l'analyse prédictive des épidémies. Cette suite logicielle se constitue d'une part de Open Data Kit (ODK) (BOKONDA, OUAZZANI-TOUHAMI, & SOUISSI, 2020), (BOKONDA, OUAZZANI-TOUHAMI, & SOUISSI, Open Data Kit: Mobile Data Collection Framework For Developing Countries., 2019) largement présenté et analysé dans les chapitres 2 et 3, ainsi que de Random Forest for Epidemiological Prediction (RF EP) d'autre part (BOKONDA, SIDIBE, SOUISSI, & OUAZZANI-TOUHAMI, 2023), présenté dans le chapitre 6, qui constitue un module externe d'analyse de données.

L'objectif que nous poursuivons avec le travail de recherche présenté dans ce chapitre de thèse est de parachever l'ajout de l'analyse prédictive dans le processus d'enquête mobile des épidémies. Nous parlons de parachever parce qu'il s'agit d'un processus que nous avons commencé depuis le chapitre 2.

En effet, il nous fallait d'abord identifier la suite logicielle la plus utilisée dans le secteur médical, dans les pays en voie de développement. Nous nous sommes, dès le début de cette thèse, posés la condition de la popularité de la suite logicielle dans les pays en voie de développement car nous souhaitons contribuer tant soit peu à l'émergence de ces nations. C'est ce que nous avons fait dans les chapitres 2 et 3. Par la suite, il était question de trouver la méthode de Machine Learning la plus adaptée pour les analyses prédictives des épidémies et l'améliorer pour mieux limiter la propagation de l'épidémie (BOKONDA, OUZZANI-TOUHAMI, & SOUISSI, Which Machine Learning method for outbreaks predictions?, 2021). Nous avons réalisé cet objectif dans les chapitres 4, 5 et 6 (BOKONDA, SIDIBE, SOUISSI, & OUAZZANI-TOUHAMI, 2023).

Ce présent chapitre peut se lire comme la fusion des résultats de deux groupes de chapitres précédemment cités, d'où sa taille plus ou moins limitées par rapport aux précédents. Il s'agit ici de montrer comment nous sommes arrivés à atteindre l'un des objectifs de cette thèse, fixé dans le chapitre 1 au travers de l'exemple empirique d'une enquête épidémiologique que nous allons dérouler sur la suite MEPS.

Avant de passer à l'enquête épidémiologique, nous allons commencer par présenter et expliquer l'architecture de la suite MEPS dans la section 2 qui sera suivie de la présentation des contraintes liées à sa mise en place notamment les prérequis ainsi que son implémentation dans la section 3. L'enquête épidémiologique, elle-même sera présentée et implémentée dans la section 4. La section 5 contient la conclusion de ce chapitre.

7.2. Architecture du MEPS

Pour une meilleure compréhension de l'architecture et du fonctionnement du MEPS, nous allons rappeler dans cette section, l'architecture d'ODK que nous avons déjà présenté dans le chapitre 3 de cette thèse. L'objectif de ce rappel est de situer le lecteur sur le schéma de l'architecture de ODK qui a pris en compte au niveau du MEPS ainsi que les outils de ODK qui y seront utilisés. Nous rappelons ainsi sur la Figure 7.1 l'architecture de ODK qui est l'exacte reprise de celle de la Figure 3.1 du chapitre 3.

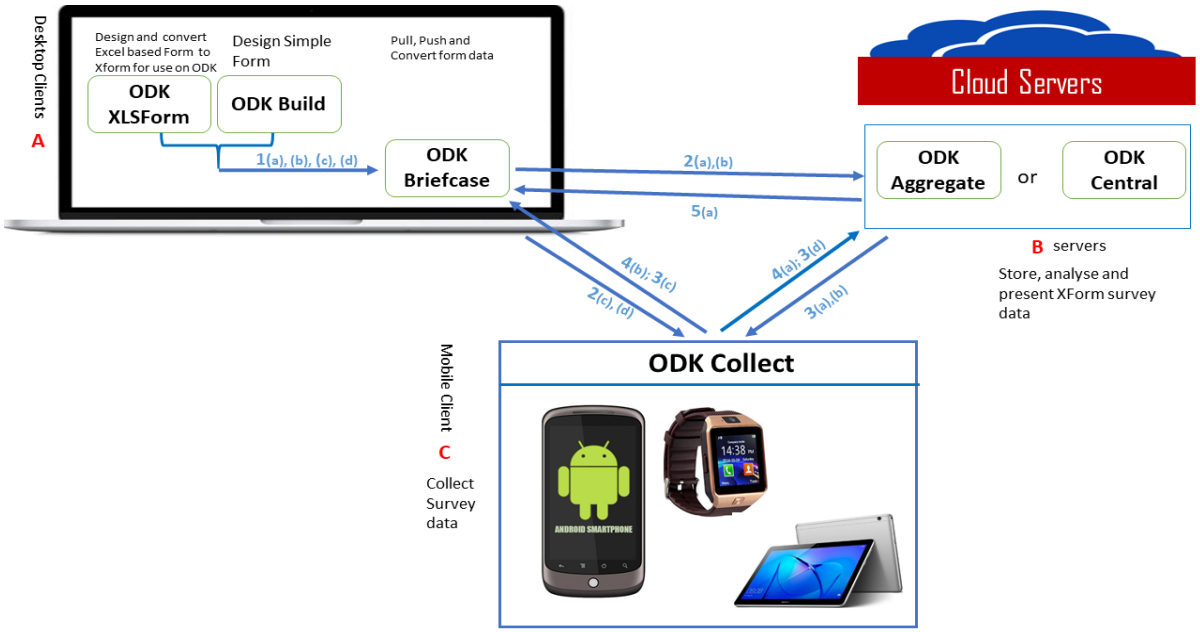


Figure 7.1: Architecture de ODK- [Source : chapitre 3]

L'explication complète de cette architecture est donnée dans la section 3.2.1 du chapitre 3 de cette thèse. Ici nous allons nous focaliser uniquement sur la trajectoire de ODK qui sera concrètement implémentée dans le cadre du MEPS. Il s'agira de la trajectoire (a). Ce qui nous conduit à la Figure 7.2 qui présente la version de l'architecture ODK utilisée dans le MEPS.

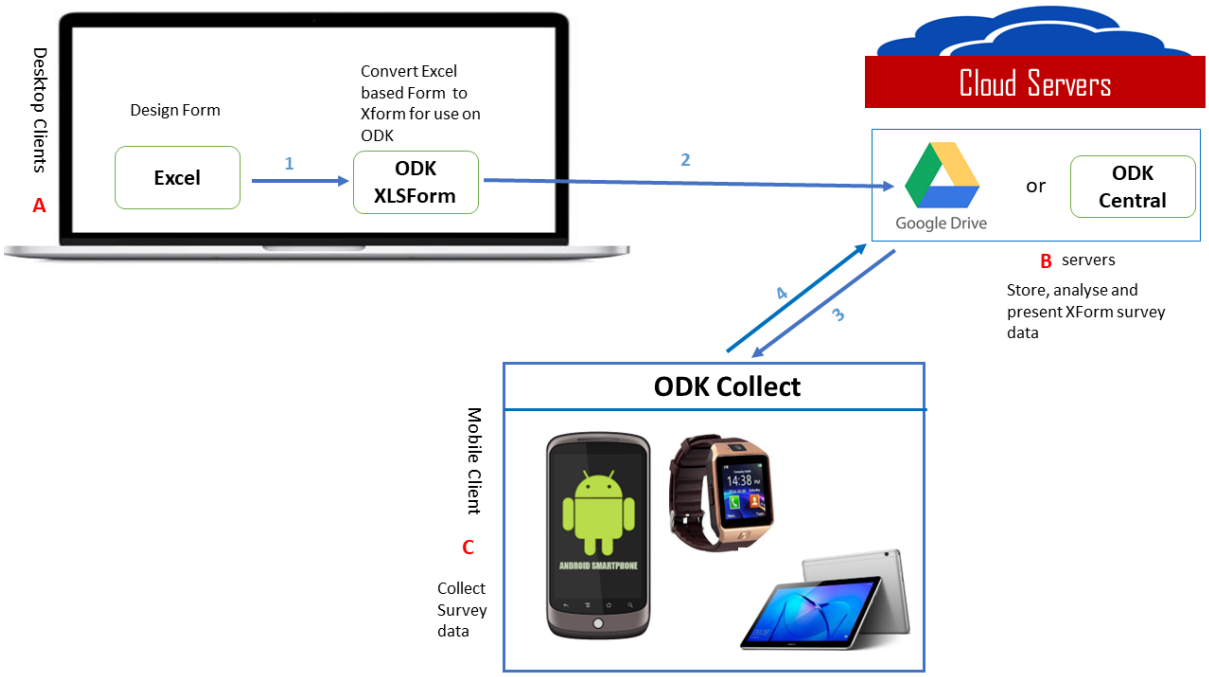


Figure 7.2 : Architecture ODK utilisée dans le système MEPS

L'architecture de ODK utilisée avec RF EP pour la mise en place du MEPS, que nous avons illustrée à la Figure 7.2 nécessite quelques explications eu égard à l'évolution de la suite logicielle ODK depuis notre première utilisation en 2020. En effet, à la date de l'écriture de cette thèse, ODK a connu plusieurs évolutions, celles qui nous intéressent dans le cadre de la mise en place du MEPS sont les suivantes :

- *L'abandon de ODK Briefcase :* Cette application de ODK qui fut autrefois utilisée comme intermédiaire entre les applications web et le serveur cloud ou alors entre les premières et ODK Collect a été supprimée de la suite. Le site officiel de ODK ne fait plus mention d'elle (ODK, 2023).
- *L'abandon de ODK Aggregate :* Bien qu'étant encore à ce jour le serveur d'ODK le plus utilisé dû à son ancienneté, ODK Aggregate n'est plus maintenu par l'équipe technique de ODK, celle-ci recommande d'utiliser ODK Central à la place d'ODK Aggregate particulièrement pour de nouveaux projets (ODK Central, 2023).
- *L'usage de Google Drive :* C'est l'option que nous avons choisi d'implémenter dans le MEPS. En effet, ODK permet maintenant d'utiliser Google Drive comme serveur cloud pour stocker les formulaires et les données collectées via ODK Collect. Contrairement à ODK Central qui est payant, Google Drive lui est gratuit (du moins les 15 premiers giga-octets), c'est ce qui justifie notre choix.

La Figure 7.2 présente le Workflow suivant :

- (1) : Le formulaire est créé avec Excel sous format xlsx puis transformé au format xml grâce à ODK XLSForm.
- (2) : Le formulaire sous le format xml est placé dans Google Drive.
- (3) : Le formulaire est téléchargé dans ODK Collect à partir de Google Drive.
- (4) : Après la collecte de données, le formulaire contenant les données est uploadé dans Google Drive sous format xlsx.

Passons maintenant à l'architecture de RF EP qui constituera le module prédiction du MEPS. Cette architecture est présentée sur la Figure 7.3. Le processus qui a conduit à la proposition de RF EP par l'enrichissement du RF classique a déjà été discuté en long et en large dans le chapitre 6 et publié dans (BOKONDA, SIDIBE, SOUISSI, & OUAZZANI-TOUHAMI, 2023). Dans ce chapitre il s'agira principalement d'utiliser RF EP dans un environnement de production en ayant ODK comme source de données ; c'est en quelque sorte une définition résumée de ce que nous avons choisi d'appeler MEPS.

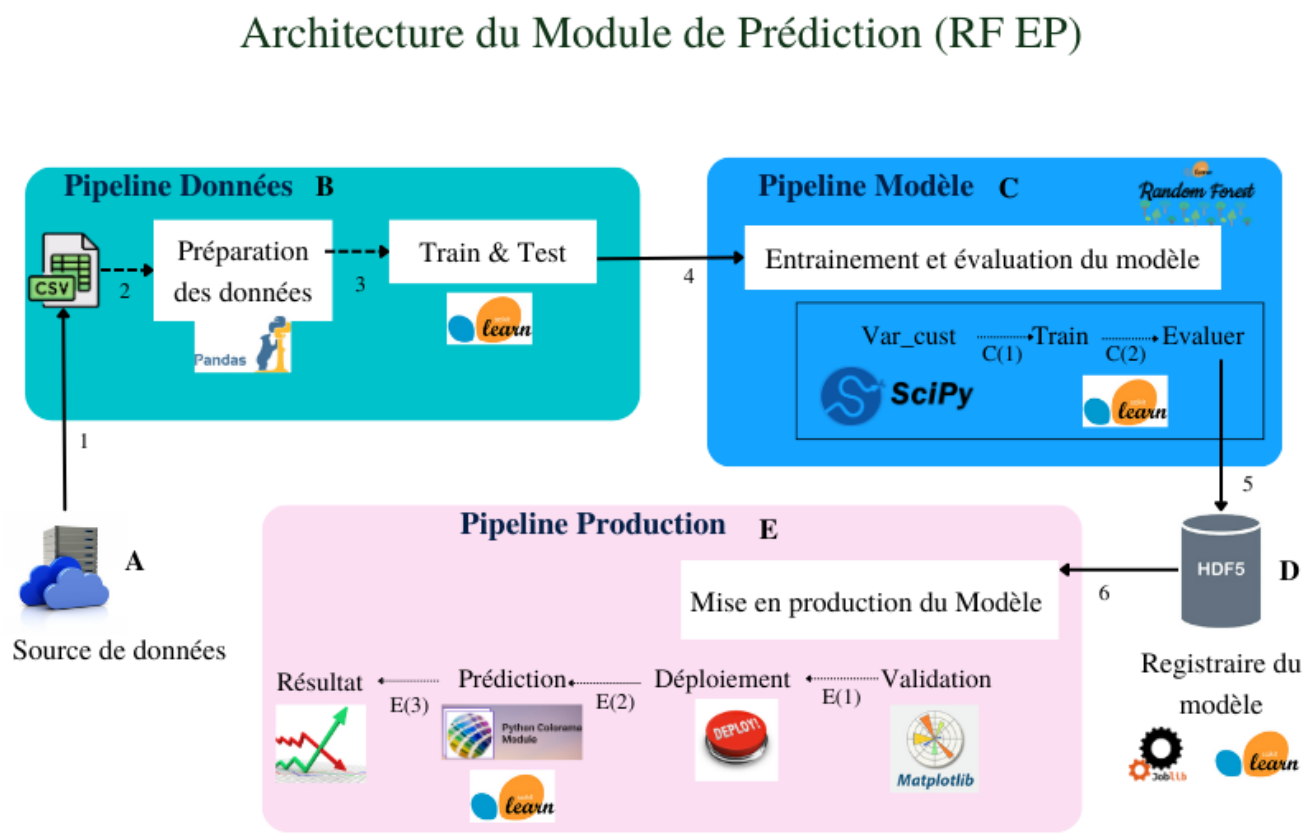


Figure 7.3 : Architecture de RF EP pour la mise en place du MEPS

Dans cette architecture de RF EP destinée à être utilisée dans le MEPS (illustrée par la Figure 7.3), nous avons ajouté à notre algorithme Forest EP toute la pile d'outils de Machine Learning nécessaires à son utilisation dans un environnement de production.

L'architecture se comporte de cinq grandes entités **A, B, C, D** et **E**. Chacune représente une structure à mettre en place :

- ***Entité A : Source de Données***

Il s'agit du système conçu pour collecter et stocker les données brutes. Celui-ci pouvant être un système mobile et interactif, notamment sur le cloud, ou bien sur un local bien déterminé. Dans ce système les données doivent déjà être structurées et suivre une certaine logique, cependant elles nécessiteront un nettoyage.

- ***Entité B : Pipeline Données***

Qui suit l'entité précédente, elle constitue l'étape acheminant les données de l'état brute à un état propre, prêtes à être utilisées pour une conception de modèle. De leur source de collecte, les données sont exportées sous format csv. Le fichier csv est chargé sur l'environnement *Jupyter Notebook* [3] (Jupyter, 2023) via le module *Pandas* (Pandas, 2023)**.**

Pandas est utilisé pour effectuer un premier aperçu des informations de nos données, une visualisation des données, quelques statistiques descriptives, tous ceci dans le but de cerner notre ensemble de données et prendre des décisions initiales sur son traitement. S'en suit une étape de nettoyage, de transformation, et de restructuration des variables et observations.

Le pré-processing des données étant effectué, les données sont scindées en deux groupes à l'aide de *Scikit-learn* (Scikit-learn, 2023).

- **Train : 80%** des données, qui seront utilisée pour entrainer le modèle
- **Test : 20%** des données, réservées pour tester le modèle et potentiellement améliorer ses paramètres.

[3] Jupyter Notebook : C'est un environnement de développement du Machine Learning qui permet d'avoir dans un seul document le code, les résultats, les graphiques, etc.

- ***Entité C : Pipeline Modèle***

Etape principale de l'étude, cette entité se sert des données mises à sa disposition, pour expliquer la variable cible. Grace à l'ensemble des autres variables explicatives, le modèle Random Forest est performé et évalué à cette étape. Trois sous étapes composent cette entité :

- **Var_cust :** Une sélection, une transformation ainsi qu'une manipulation de l'espace des variables de façon totalement supervisée. Il s'agit d'un **feature_selection**, qui nous permet de choisir quelles variables sont importantes dans la conception du modèle. La fonction *SelectKBest* (KBest in Scikit, 2007) avec le critère de *Chi2* (Gajawada, 2019) et un seuil, tous appartenant au package *Scikit-learn* (Scikit-learn, 2023), nous ont permis d'effectuer cette tâche.

  Par la suite, une normalisation (**scaling**) des données est effectuée, car le poids de certaines variables (c'est-à-dire la mesure ou valeur de celle-ci. Plus une variable contient des mesures élevées, plus elle sera caractérisée de poids élevé et l'écart entre cette variable avec les autres, pourrait affecter le processus de calculs), comme la température et l'âges, pourraient impacter la modélisation.

  Et pour finir une réduction de dimension (**PCA** pour Principal Component Analysis), qui nous permet de remodéliser l'ensemble des données pour le confiner dans un nouveau jeu avec moins de variables. Ensuite une décision sur le nombre de composantes principales est prise, en essayant de maximiser le pourcentage de variabilité expliquée.

  A la sortie de cette étape nous avons un jeu de données plus organisé, restructuré et moins dense.
- **Entraînement du modèle :** à ce niveau, le modèle est entraîné en utilisant les données préparées à l'étape précédente. Cela peut prendre plusieurs itérations, en ajustant les différents hyperparamètres[4] du modèle pour trouver les meilleures performances. Des validations croisées sont utilisées pour optimiser lesdits hyperparamètres afin de rendre meilleur le modèle. **Scipy** (SciPy, 2023) intervient dans cette phase pour l'optimisation.

  Évaluation du modèle : une fois le modèle entraîné, il est important de le tester sur de nouvelles données pour évaluer sa performance. Cela permet de déterminer si le modèle

[4] Hyperparamètres : qui sont des paramètres externes, utilisés pour ajuster le modèle d'apprentissage automatique avant l'entraînement. Contrairement aux paramètres du modèle, qui sont appris par le modèle lors de l'entraînement, les hyperparamètres sont définis avant l'entraînement et déterminent la façon dont le modèle apprend.

est généralisable et s'il peut être utilisé en production. D'où l'utilité du jeu Test. *Scikit-learn* fournit des techniques et métriques optimales d'évaluation de modèle, telles que celles traitées dans le chapitre 6 de cette thèse (la matrice de confusion, accuracy, precision, le recall et le f1-score).

- ***Entité D : Registre du Modèle***

Une sorte de check-point où le modèle est mémorisé, afin d'être utilisé plus tard. Un registre de modèles est une base de données ou un système utilisé pour stocker et gérer les modèles d'apprentissage automatique (Soh, Singh, Soh, & Singh, 2020). Il permet de suivre les différents modèles qui ont été entraînés et déployés et de stocker des métadonnées sur chaque modèle, telles que la version, les données d'entraînement utilisées, les métriques d'évaluation et toutes les notes ou commentaires pertinents. Les registres de modèles sont souvent utilisés dans les flux de travail d'apprentissage automatique pour permettre la collaboration, le contrôle de version et la reproductibilité (Coursera, 2023). Ils peuvent également être utilisés pour stocker des modèles dans le serveur ou pour le déploiement et pour permettre un accès facile aux modèles pour le test et l'évaluation futurs (Coursera, 2023). Des fonctions sur *scikit-learn* sont déjà disponibles pour réaliser cet enregistrement.

- ***Entité E : Pipeline Production***

Notre modèle RF EP ayant été entraîné et testé peut-être mis en production. RF EP est déployé pour ensuite être utilisé et effectuer des prédictions sur de nouvelles données au sein de notre environnement de production. L'environnement de production est utilisé pour automatiser le processus de prise de décision (de prédiction) sur la maladie. En général, pour être utilisés en production, les modèles de Machine Learning doivent être robustes, fiables et performants, et ils doivent être mis en œuvre de manière à être facilement maintenus et mis à jour au besoin. C'est ce qu'assure cette entité.

Il est important de suivre la performance du modèle une fois qu'il est en production, et de le mettre à jour si nécessaire pour maintenir ses performances. MLFlow fournit des outils pour suivre l'ensemble de ce cycle de vie, en enregistrant les différents *runs d'expérimentation et* en permettant de déployer facilement le modèle en production. Les modules *scikit-learn* et *colorama* (Helskyaho, et al., 2021) sont intervenus pour effectuer les tâches de cette entité.

La sortie de cette entité est le résultat final du système MEPS. Il s'agit pour un nouvel individu, la probabilité d'être positif à une épidémie, ladite probabilité est donnée dans l'intervalle [0-1].

Le workflow de l'architecture de RF EP présentée à la Figure 7.3 doit être compris de la manière suivante :

- (1) : Premier flux, qui chemine les données depuis leur source, vers l'entité de traitement des données, sous format csv.
- (2) : Les données sont chargées sur l'environnement notebook afin de subir une préparation. Il s'agit ici de s'assurer que les données sont dans le bon format pour être comprises et utilisées par le modèle Machine Learning (Random Forest). C'est à cette étape que s'appliquera par exemple le dummy encodage évoqué à la section 6.3.2 du chapitre 6 de cette thèse.
- (3) : après la préparation de données, vient l'étape où elles seront subdivisées en deux jeux de données. Un jeu de données pour l'entrainement du modèle (Train) et un deuxième pour le test (Test).
- (4) : ensuite vient l'étape de l'entrainement et de l'évaluation du modèle sur la base de données déjà subdivisées en deux datasets. Au sein du Pipeline Modèle s'exécute les sous-étapes suivantes, dans cet ordre :
    - ✓ C (1) : l'algorithme Var_Cust qui fait partie de l'algorithme Forest EP est appliquées aux données. Les étapes de cet algorithme sont largement discutées dans la section 6.3.3 du chapitre 6 de cette thèse ainsi que dans les lignes qui précèdent . Grosso modo, Var_Cust se compose de trois étapes : la sélection des variables significatives, la normalisation de données et la réduction des dimensions.
    - ✓ C (2) : le modèle est par la suite entrainé puis évalué. Il faudra noter que l'évaluation du modèle est faite notamment grâce aux métriques. Il s'agit des métriques traitées dans la section 6.3.3 du chapitre 6 de cette thèse.
- (5) : La version optimale du modèle est enregistrée sur une base de modèle en vue d'être réutilisée plus tard.
- (6) : Il s'agit de la dernière étape, la mise en production du modèle. Elle consiste à utiliser le modèle dans un environnement de travail afin d'obtenir le résultat au sujet d'un patient. Cette étape de mise en production est constituée des sous-étapes suivantes :
    - ✓ E (1) : Après être validé par une expertise (des techniques de validation de modèle dans les précédentes étapes), E (1) assure le déploiement du modèle.

- ✓ E (2) : Le modèle déployé est destiné à prédire pour un nouvel individu.
- ✓ E (3) : La fin du workflow est marquée par l'affichage du résultat.

Nous pouvons constater que sur l'architecture de RF EP l'entité A est une source de données, comme dit ci-haut. C'est à ce niveau que viendra se greffer ODK pour former le MEPS. D'où la Figure 7.4 qui illustre l'architecture du MEPS. Cette architecture n'a besoin d'aucune explication supplémentaire que celles qui ont été données précédemment et séparément sur ODK et RF EP. La section suivante (contrainte et implémentation) permettra d'avoir une idée beaucoup plus concrète de cette architecture.

Bien que toute l'explication de l'architecture du MEPS présentée sur la Figure 7.4 soit contenue dans celles des architectures de ODK et RF EP données précédemment, il sied tout de même de souligner que dans le système MEPS le dataset sera continuellement alimenté par les données collectées via ODK et stockées dans un fichier xlsx (ou csv) dans Google Drive.

En effet, grâce à une synchronisation entre ODK Collect et Google Drive le fichier xlsx (ou csv) sera alimenté à chaque fois que l'utilisateur de ODK Collect va collecter les données et les envoyer au serveur. Et c'est le même fichier qui sera utilisé par RF EP pour effectuer la prédiction.

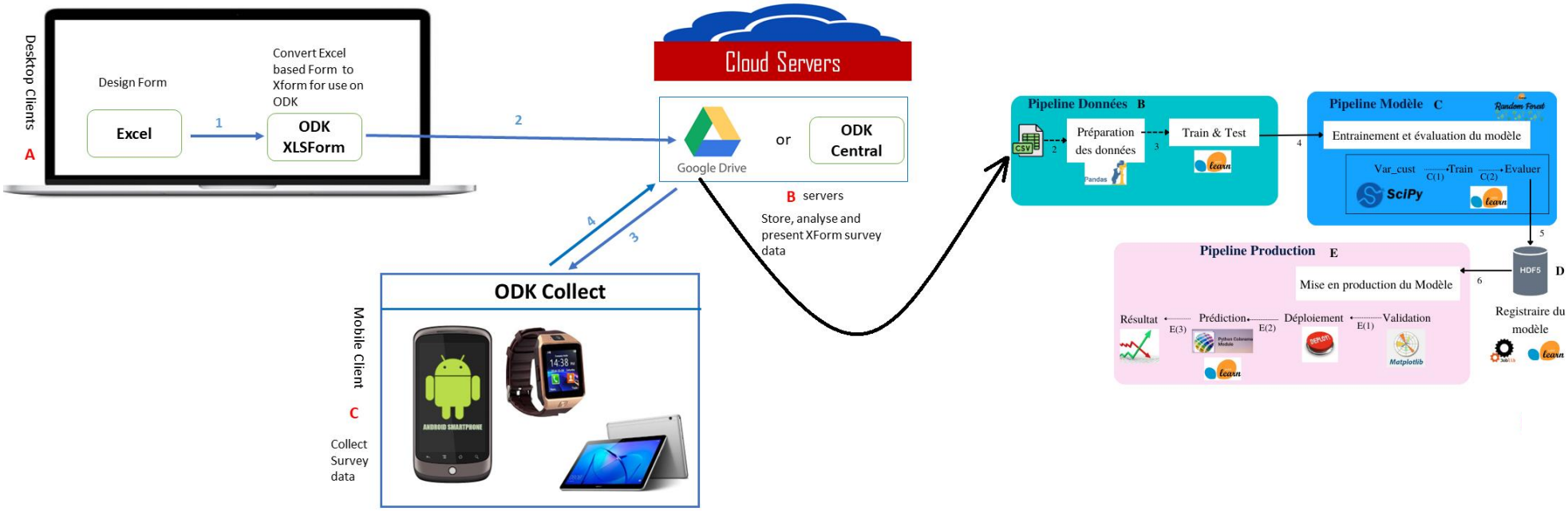


Figure 7.4 : Architecture du MEPS

7.3. Contraintes et Implémentation du MEPS

En ce qui concerne les contraintes dans l'implémentation et/ou l'utilisation du MEPS, elles ne sont autres que celles liées à ses deux composantes. Premièrement ODK Collect ne peut être installé que sur un système d'exploitation mobile Android. Il faudra donc que la personne qui désire utiliser le MEPS puisse posséder un tel système. A la date de rédaction de cette thèse, ODK Collect n'est pas encore disponible pour les appareils iOS. Ceci s'explique par le fait qu'ODK a été spécialement conçu pour être utilisé dans les pays en voie de développement comme nous l'avons détaillé dans les chapitres 2 et 3 de cette thèse. Et nous savons que dans lesdits pays le pouvoir d'achat de la population est assez limité, d'où le choix des appareils Android. Pour les utilisateurs d'Android, il suffira d'aller dans la Play Store et télécharger ODK Collect comme illustré sur la Figure 7.5, dans le cadre d'une simulation.

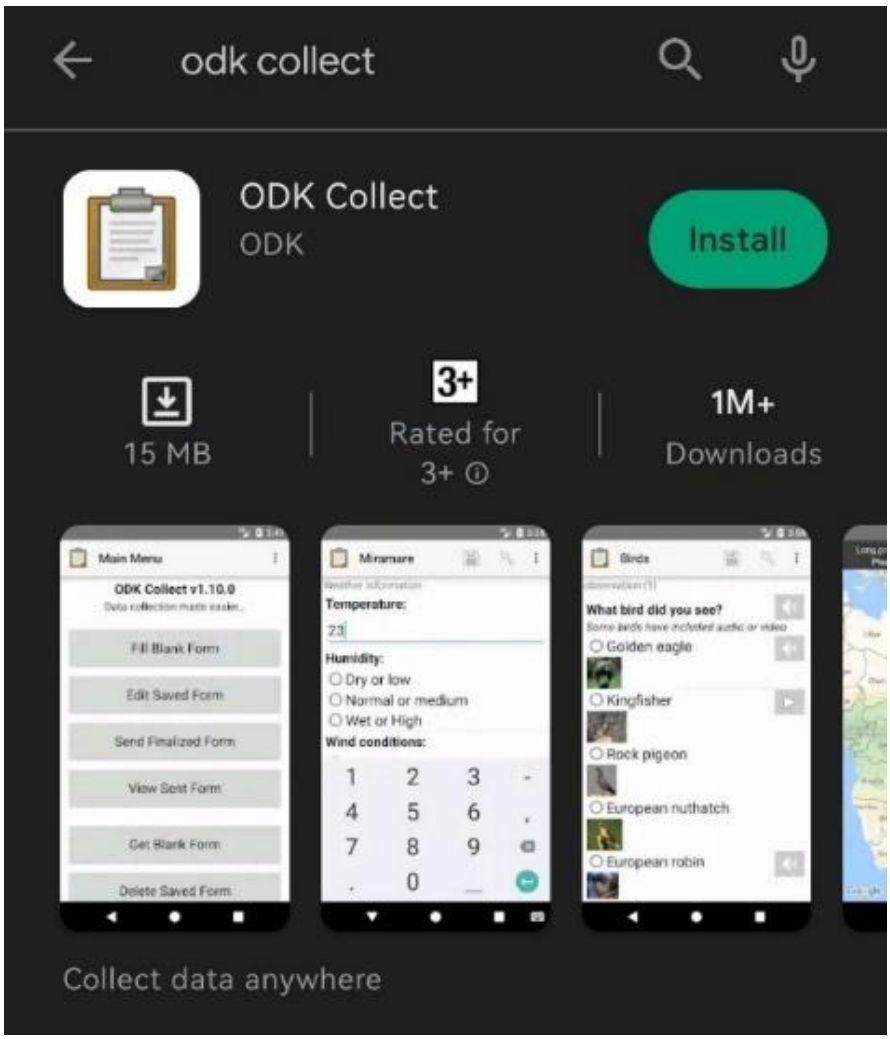


Figure 7.5 : Téléchargement de ODK Collect dans la Play Store

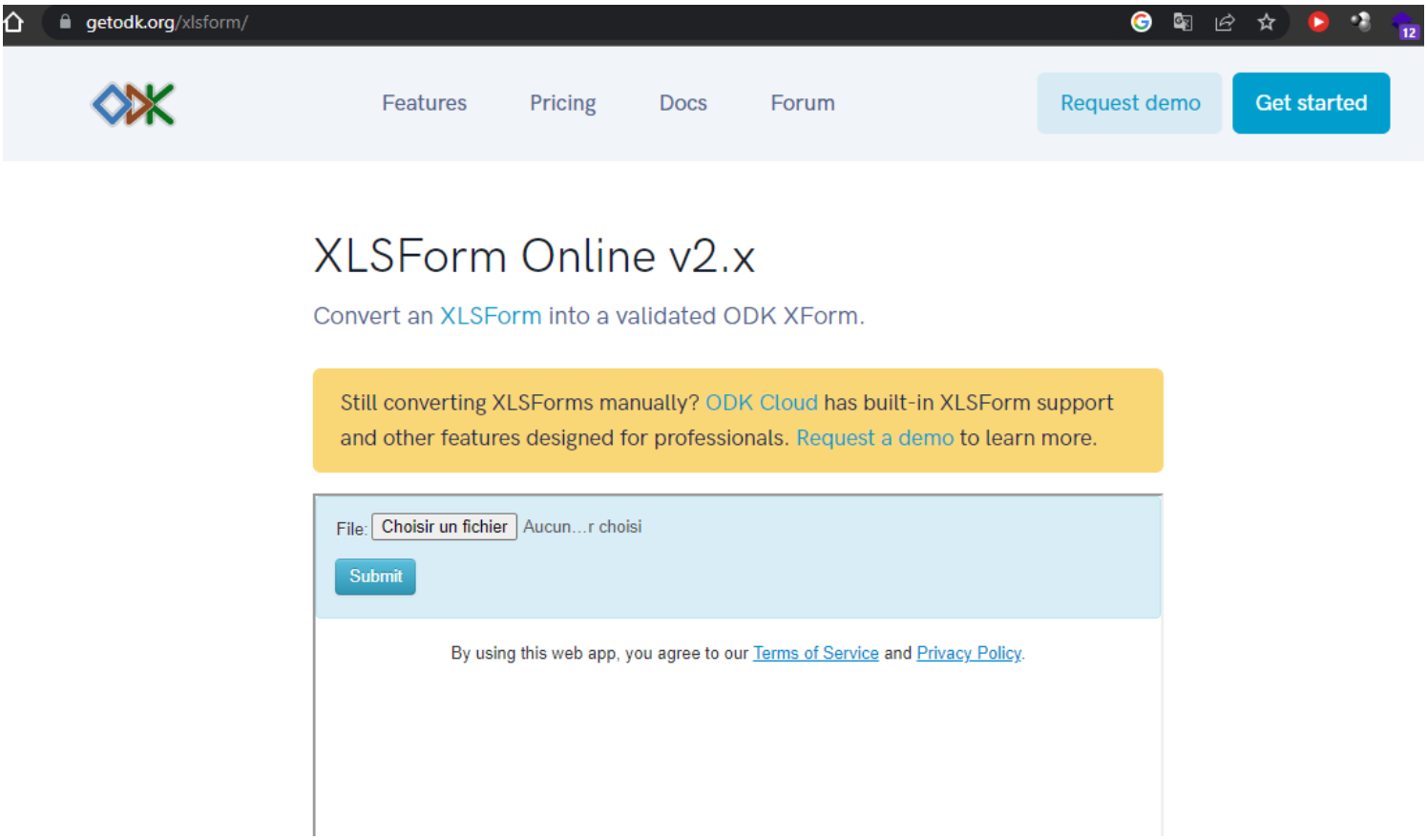


Figure 7.6 : ODK XLS Form en ligne

Quant à l'utilisation de Google Drive comme serveur, il faudra naturellement posséder ou créer une adresse électronique Gmail qui donne droit automatiquement à 15 giga-octets d'espace cloud de stockage, pour le besoin d'implémentation du cas d'utilisation qui sera présenté dans la section suivante nous avons créé l'adresse suivante : driveforrfep@gmail.com. Les autres outils ODK ne présentent aucune contrainte particulière. ODK XLSForm est accessible en ligne (XLSForm Online, 2023) et facile d'usage comme le montre la Figure 7.6. Il suffit d'y accéder en ligne et d'y uploader le fichier xlsx (ou csv) pour que celui-ci soit converti en xml. Pour une meilleure adaptation du format du formulaire Excel, un template a été mis en place par l'équipe de ODK et est publiquement accessible (XLSForm template, 2023).

Quant à la partie analyse de données avec RF EP, la manière la plus adaptée de posséder la quasi-totalité des librairies et outils nécessaires est d'installer directement la distribution Anaconda (Anaconda, 2023). Il s'agit d'une distribution très connue par les data scientistes mais elle a pour seul inconvénient d'être assez lourde. Il faudra donc posséder un ordinateur performant, de préférence 8 Go de RAM, au moins. Notre implémentation a été effectué sur un

ordinateur avec 12 Go de RAM. Une fois Anaconda installée, son icône ainsi que sa console paraitront sur le menu démarrer comme le montre la Figure 7.7.

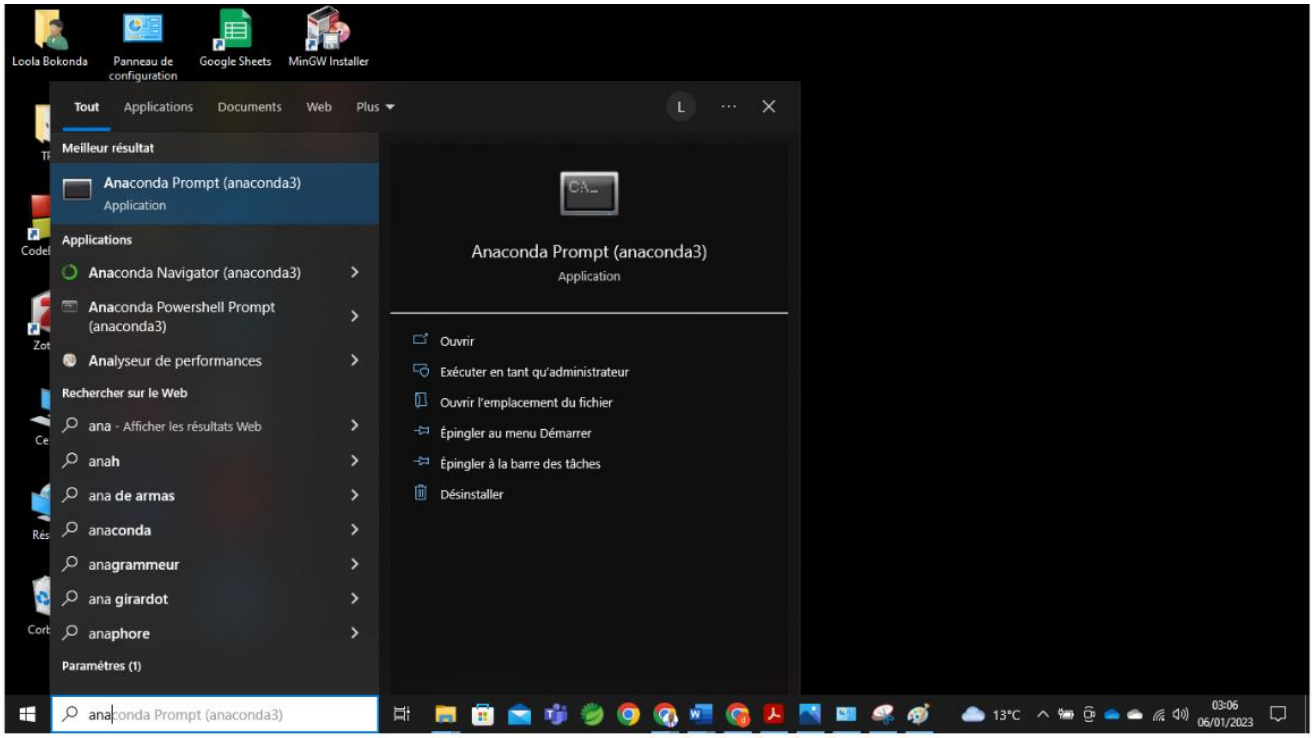


Figure 7.7 : La distribution Anaconda et sa console

La section suivante complétera l'implémentation du MEPS au travers d'un cas d'utilisation, par simulation d'une enquête mobile épidémiologique que nous avons menée.

7.4. Cas d'utilisation par simulation

Le cas d'utilisation que nous allons exposer dans cette section correspond au dataset décrit à la section 6.2.2.1 du chapitre 6 de cette thèse. Le but de l'étude est de parvenir à prédire si une personne est potentiellement positive à la covid-19 à partir de dix variables cliniques. Le modèle RF EP est préalablement entrainé et testé grâce audit jeu de données suivant le processus détaillé dans le chapitre 6, et qui correspond aux entités B & C de l'architecture de RF EP présenté sur la Figure 7.3.

Nous avons donc formulé notre enquête sous forme de dix questions (excluant la variable cible, le but étant de retrouvée cette dernière) afin de récupérer des données bien précises selon l'encodage du dataset. Le Tableau 7.1 résume notre enquête en termes des questions directement posées à l'utilisateur, des réponses possibles à ces questions et des valeurs récupérées par le système.

Tableau 7.1 : Questionnaire de l'enquête mobile épidémiologique par simulation

Question	Valeurs utilisateur	Valeur système
Quel est votre âge ?	[0-120]	[0-120]
Quelle est votre température (entre 98-104°F) ?	[98-104]	[98-104]
Souffrez-vous de douleur dans le corps ?	(Oui/non)	(1/0)
Votre nez coule-t-il ?	(Oui/non)	(1/0)
Souffrez-vous de problème respiratoire ?	(Oui/non)	(1/0)
Souffrez-vous de congestion nasale ?	(Oui/non)	(1/0)
Souffrez-vous de douleur à la gorge ?	(Oui/non)	(1/0)
Votre genre ?	(Masculin/féminin)	(0/1)
Quel est votre degré de la maladie ?	(Modéré/moyen/sévère)	(0/1/2)
Avez-vous été en contact avec un patient ayant la covid ?	(Non/J'ignore/Oui)	(0/1/2)

Ces dix questions cliniques sont collectées au travers de l'application ODK Collect de la suite logicielle ODK. Cependant, avant de procéder à la collecte de données via ODK Collect, il faudra au préalable configurer les paramètres du serveur de ODK Collect pour que celui-ci pointe vers l'instance de Google Drive souhaitée. Cette configuration, pour ce cas d'utilisation est illustrée par la Figure 7.8.

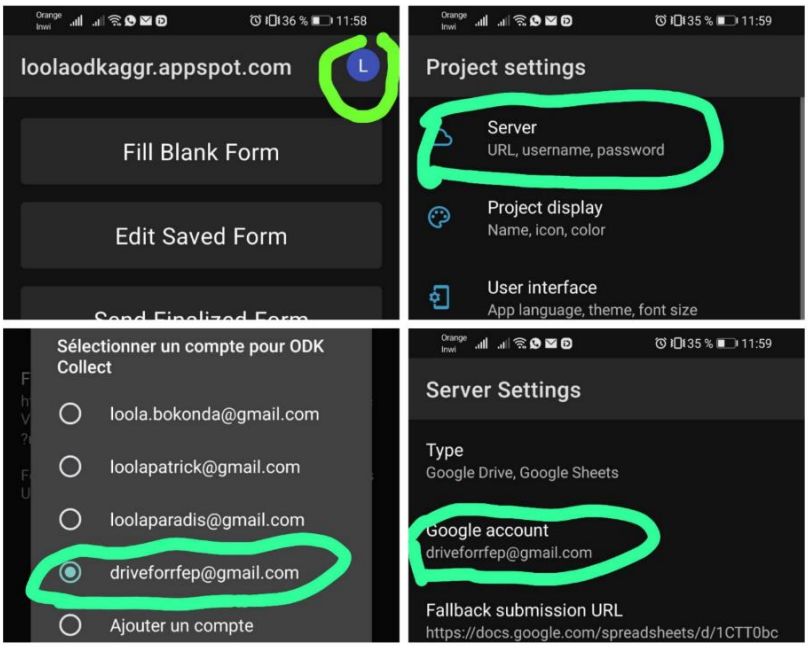


Figure 7.8 : Configuration de Google Drive comme serveur ODK Collect

Après avoir configuré le serveur de ODK Collect, nous avons procédé au téléchargement du formulaire que nous avions pris le soin de placer sur l'instance Google Drive liée à l'adresse électronique driveforrfep@gmail.com. En effet, une fois ce formulaire téléchargé sur l'application mobile ODK Collect, il est visible dans l'onglet '*Fill Blank Form'* de celle-ci, comme le montre la Figure 7. 9.

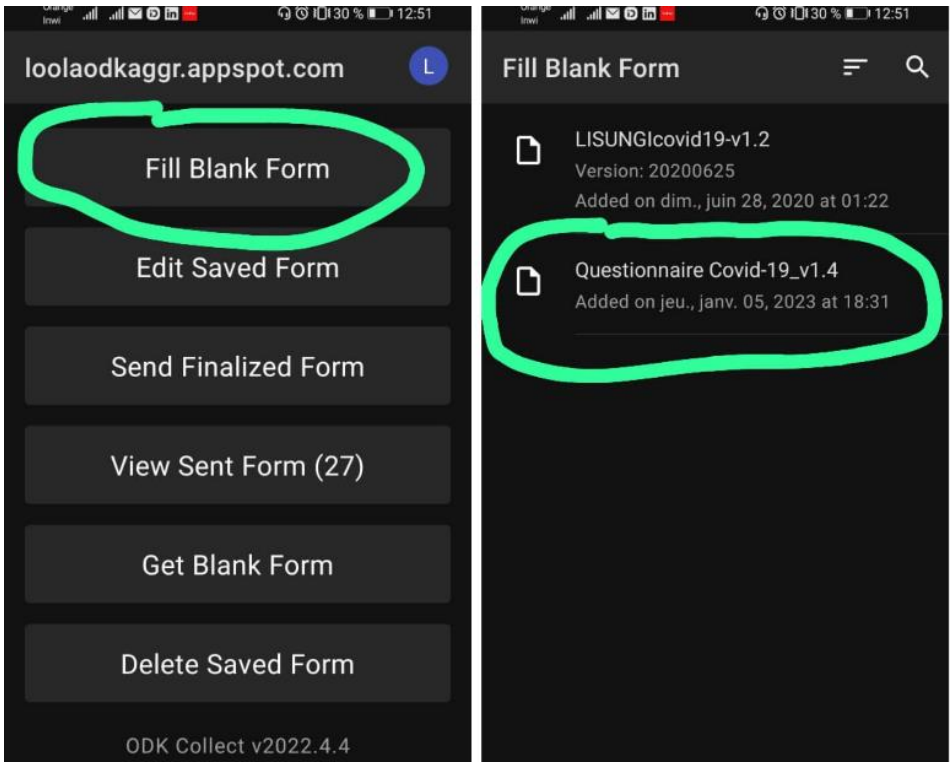


Figure 7.9 : Formulaire disponible sur ODK Collect

C'est à la suite de ce téléchargement que nous avons commencé notre simulation. La simulation en elle-même est constituée de deux étapes :

a) Collecte de données via ODK Collect.
b) Obtention des résultats via RF EP.

Néanmoins, il sied de souligner qu'avant ces deux étapes, notre algorithme a été entrainé par des données similaires, et qu'entre les deux étapes, les données collectées par ODK Collect et uploadées sur Google Drive sont téléchargées sous format csv pour être utilisées sur Jupyter Notebook.

Les Figures 7. 10 et 7.11 illustrent un exemple de données d'un utilisateur collectées via ODK Collect.

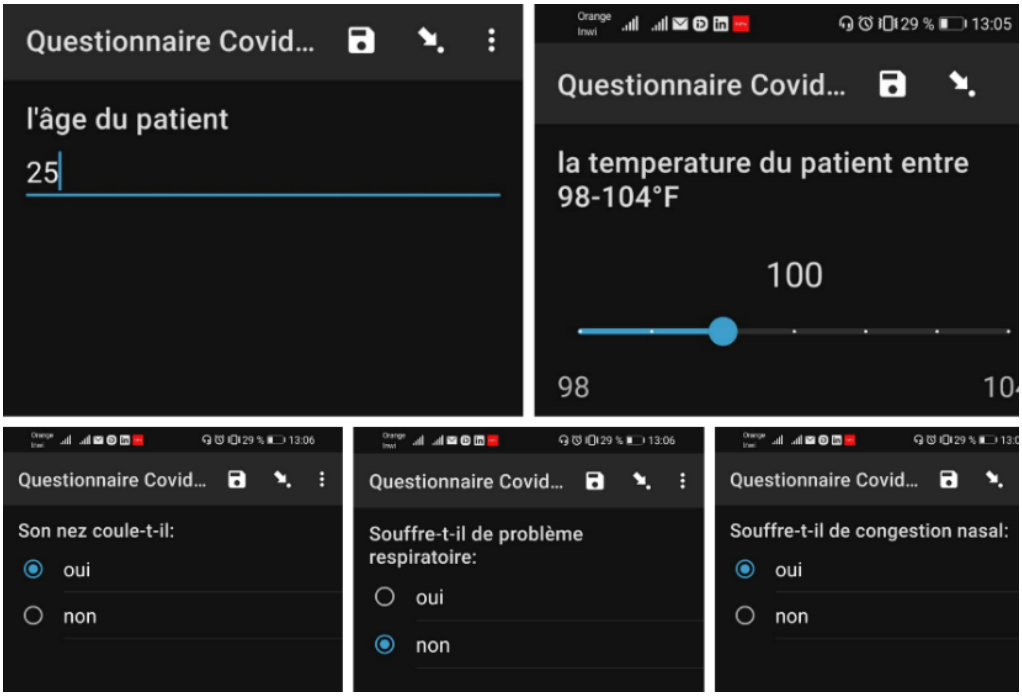


Figure 7.10 : Cas d'utilisation-Exemple-Collecte de données-Partie 1

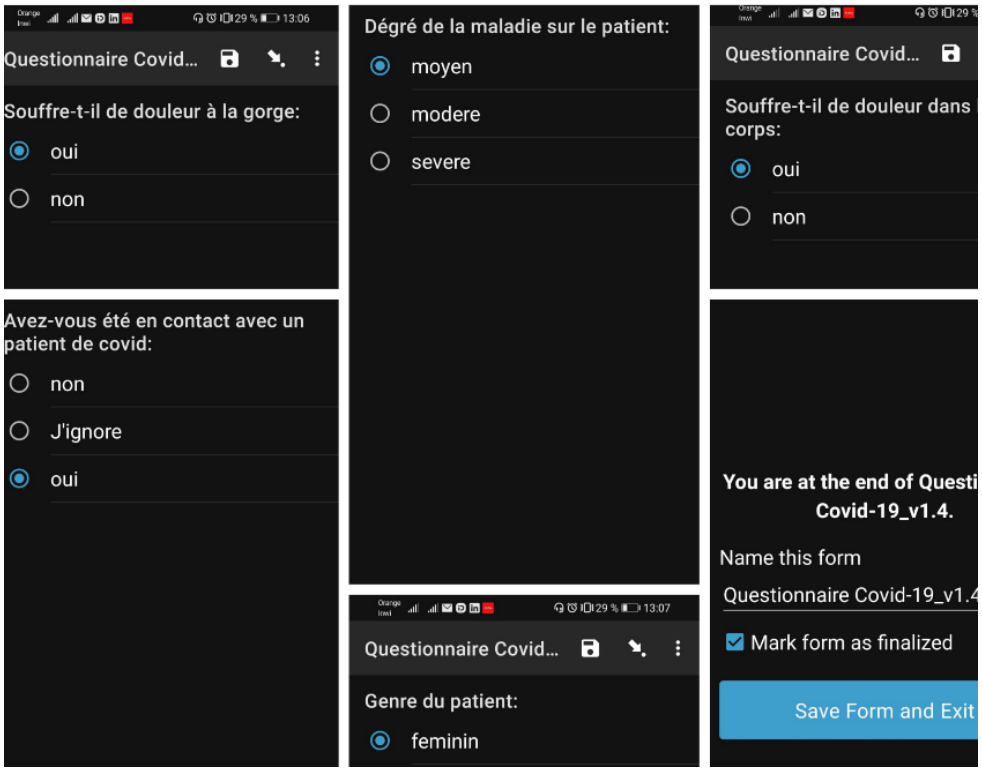


Figure 7.11 : Cas d'utilisation-Exemple-Collecte de données-Partie 2

Nous pouvons par la suite observer, sur la Figure 7.12 le résultat obtenu grâce à RF EP pour ledit patient. Le verdict final est que le patient est très probablement positif à la covid-19 avec une probabilité de 0.70 (celle-ci étant calculée par le modèle entrainé au préalable).

```
Entrée [27]: predict(X_train_s,X_test_s,y_train,randomclassi)

             Entrez l'âge du patient:25
             Entrez la temperature du patient entre 98-104 :100
             Souffre-t-il de douleur dans le corps(oui/non)-(1/0):1
             Son nez coule-t-il(oui/non)-(1/0):1
             Souffre-t-il de problème respiratoire(oui/non)-(1/0):0
             Souffre-t-il de congestion nasal (oui/non)-(1/0):1
             Souffre-t-il de douleur à la gorge (oui/non)-(1/0):1
             Genre du patiente(Masculin/féminin)-(0/1):1
             Dégré de la maladie sur le patient(modere/moyen/severe)-(0/1/2):1
             Avez-vous été en contact avec un patient de covid (Non/J'ignore/Oui)-(0/1/2):2
             Les variables les plus significatives retenues après la selection des variables sont:  ['Age', 'Bodyp
             ain', 'Runny_nose', 'Difficulty_in_breathing', 'Nasal_congestion', 'Sore_throat', 'Severity_Mild', 'S
             everity_Moderate', 'Severity_Severe', 'Contact_with_covid_patient_no', 'Contact_with_covid_patient_no
             t known', 'Contact_with_covid_patient_yes']
             La probabilité que le patient soit positive au covid est:0.7009101760056689
             Malheureusement le patient est considéré positive au covid 😒😒😒
```

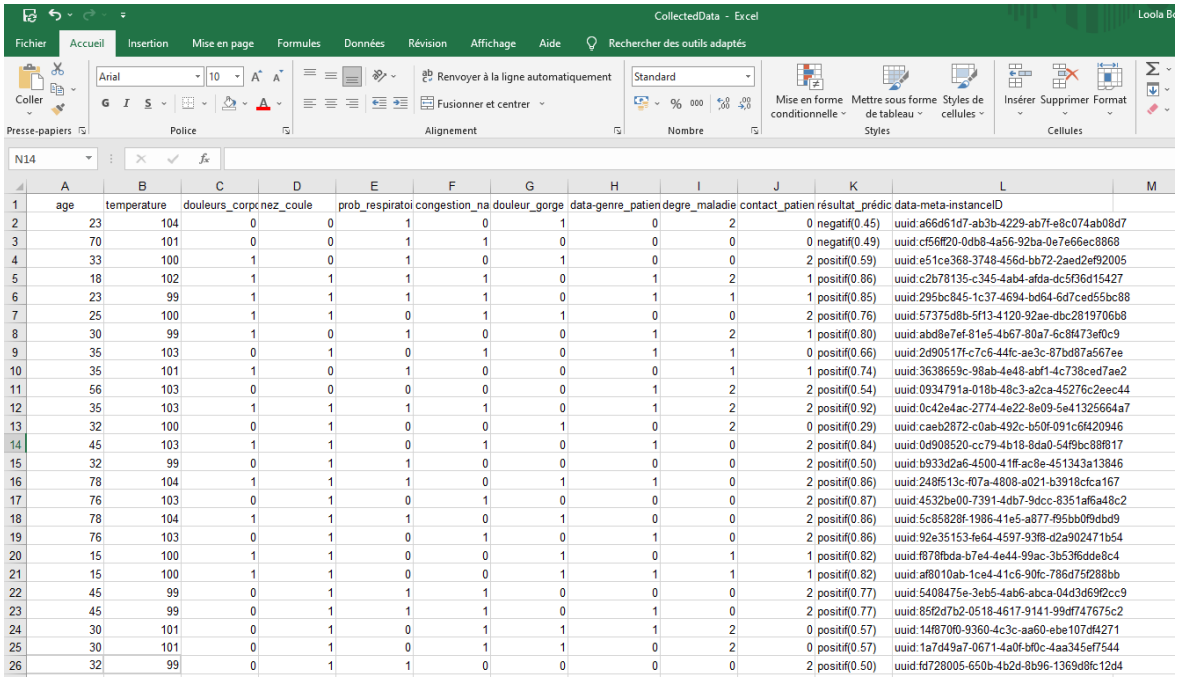
Figure 7.12: Résultat de la prédiction de l'état du patient par RF EP

Outre le cas personnel de l'individu présenté dans les précédentes lignes, cette étude de cas a porté sur vingt-cinq nouvelles personnes. La Figure 7.13 donne la liste de toutes les vingt-cinq personnes en termes des caractéristiques et résultats de prédiction de l'algorithme.

Parmi ces personnes deux seulement ont été prédites négatives et vingt-trois positives. Pour observer son comportement et le résultat qui en découlerait, nous avons exprès donné à l'algorithme les dix dernières personnes avec exactement les mêmes informations à l'exception du genre (autrement dit cinq femmes et cinq hommes avec exactement les mêmes caractéristiques). Il en résulte que dans la totalité des cas la prédiction est la même, comme le montre la Figure 7.13 sur les dix dernières lignes. Nous pouvons donc en conclure que pour notre algorithme le genre n'est pas un facteur déterminant. On ne peut pas dire, se basant sur le résultat de notre algorithme, que les hommes sont plus susceptibles d'être contaminés que les femmes ou l'inverse.

Le seuil est fixé à 0.5 pour rappel. A partir et au-delà de cette valeur, le patient est considéré positif. Pour être certain de la contamination, cette probabilité doit s'avoisiner à 1 et de 0 pour être considérer comme non malade. Ainsi pour une valeur aux alentours de 50, il est difficile de s'affirmer, le mieux serait de préserver ces cas, à l'expertise médicale.

La colonne « data-meta-instanceID » est une anonymisation des données, utile pour permettre de garder sécurisés les données des utilisateurs ainsi que le résultat de la prédiction. Ce sont des identifiants qui permettraient de retrouver, aussi facilement, le patient.



CollectedData - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	age	temperature	douleurs_corp	nez_coule	prob_respiratoi	congestion_na	douleur_gorge	data-genre_patien	degre_maladie	contact_patien	résultat_prédic	data-meta-instanceID	
2	23	104	0	0	1	0	1	0	2	0	negatif(0.45)	uuid:a66d61d7-ab3b-4229-ab7f-e8c074ab08d7	
3	70	101	0	0	1	1	0	0	0	0	negatif(0.49)	uuid:cf56ff20-0db8-4a56-92ba-0e7e66ec8868	
4	33	100	1	0	1	0	1	0	0	2	positif(0.59)	uuid:e51ce368-3748-456d-bb72-2aed2ef92005	
5	18	102	1	1	1	1	0	1	2	1	positif(0.86)	uuid:c2b78135-c345-4ab4-afda-dc5f36d15427	
6	23	99	1	1	1	1	0	1	1	1	positif(0.85)	uuid:295bc845-1c37-4694-bd64-6d7ced55bc88	
7	25	100	1	1	0	1	1	0	0	2	positif(0.76)	uuid:57375d8b-5f13-4120-92ae-dbc2819706b8	
8	30	99	1	0	1	0	0	1	2	1	positif(0.80)	uuid:abd8e7ef-81e5-4b67-80a7-6c8f473ef0c9	
9	35	103	0	1	0	1	0	1	1	0	positif(0.66)	uuid:2d90517f-c7c6-44fc-ae3c-87bd87a567ee	
10	35	101	1	0	1	0	0	0	1	1	positif(0.74)	uuid:3638659c-98ab-4e48-abf1-4c738ced7ae2	
11	56	103	0	0	0	0	0	1	2	2	positif(0.54)	uuid:0934791a-018b-48c3-a2ca-45276c2eec44	
12	35	103	1	1	1	1	0	1	2	2	positif(0.92)	uuid:0c42e4ac-2774-4e22-8e09-5e41325664a7	
13	32	100	0	1	0	0	1	0	2	0	positif(0.29)	uuid:caeb2872-c0ab-492c-b50f-091c6f420946	
14	45	103	1	1	0	1	0	1	0	2	positif(0.84)	uuid:0d908520-cc79-4b18-8da0-54f9bc88f817	
15	32	99	0	1	1	0	0	0	0	2	positif(0.50)	uuid:b933d2a6-4500-41ff-ac8e-451343a13846	
16	78	104	1	1	1	0	1	1	0	2	positif(0.86)	uuid:248f513c-f07a-4808-a021-b3918cfca167	
17	76	103	0	1	0	1	0	0	0	2	positif(0.87)	uuid:4532be00-7391-4db7-9dcc-8351af6a48c2	
18	78	104	1	1	1	0	1	0	0	2	positif(0.86)	uuid:5c85828f-1986-41e5-a877-f95bb0f9dbd9	
19	76	103	0	1	0	1	0	1	0	2	positif(0.86)	uuid:92e35153-fe64-4597-93f8-d2a902471b54	
20	15	100	1	1	0	0	1	0	1	1	positif(0.82)	uuid:f878fbda-b7e4-4e44-99ac-3b53f6dde8c4	
21	15	100	1	1	0	0	1	1	1	1	positif(0.82)	uuid:af8010ab-1ce4-41c6-90fc-786d75f288bb	
22	45	99	0	1	1	1	0	0	0	2	positif(0.77)	uuid:5408475e-3eb5-4ab6-abca-04d3d69f2cc9	
23	45	99	0	1	1	1	0	0	1	2	positif(0.77)	uuid:85f2d7b2-0518-4617-9141-99df747675c2	
24	30	101	0	1	0	1	1	1	2	0	positif(0.57)	uuid:14f870f0-9360-4c3c-aa60-ebe107df4271	
25	30	101	0	1	0	1	1	0	2	0	positif(0.57)	uuid:1a7d49a7-0671-4a0f-bf0c-4aa345ef7544	
26	32	99	0	1	1	0	0	0	0	2	positif(0.50)	uuid:fd728005-650b-4b2d-8b96-1369d8fc12d4	

Figure 7.13 : Caractéristiques et résultats de l'étude de cas

7.5. Conclusion

Tout bien considéré, ce chapitre a porté sur une première version du système mobile de prédiction épidémiologique. Il s'agit, en effet, de l'ébauche d'une suite logicielle qui a pour ambition d'être une extension de ODK spécialement conçue pour la prédiction des épidémies dans les pays en voie de développement.

En effet, l'architecture présentée à la Figure 7.4 ainsi que l'implémentation et le cas d'utilisation qui ont suivi posent les jalons de la suite logicielle que nous appelons de nos vœux. Nous souhaitons construire une suite logicielle qui partira de ODK, se servant de tous les avantages qu'offre cette plateforme y compris sa popularité, et ajoutera à celle-ci un module d'analyse prédictive. Le module d'analyse prédictive que nous utilisons ici n'est rien d'autre que RF EP dont la conception et l'implémentation ont été publié dans (BOKONDA, SIDIBE, SOUISSI, & OUAZZANI-TOUHAMI, 2023) et présenté dans le chapitre 6 de ce mémoire de thèse. Le MEPS peut donc, légitimement et dans une perspective d'être amélioré dans le futur, se résumer par la formule : MEPS = ODK+RF EP.

Chapitre 8 : Conclusion et Perspectives

8.1. Conclusion

Somme toute, les méfaits des épidémies dans le monde ne sont plus à démontrer. Entre morts précoces, paralysies, et handicaps, les épidémies sont des fléaux qui impactent profondément les sociétés frappées par elles. Qu'il s'agisse des pays développés ou ceux qui sont en cours de développement, aucun ne peut rester indifférent à l'apparition ou la résurgence d'une épidémie. Ainsi, la lutte contre les épidémies est une problématique universelle qui ne peut être ignorée au vu de ses conséquences indéniables.

Nul ne peut réfuter que dans la lutte contre les épidémies, les enquêtes s'avèrent un outil indispensable, pourvu qu'elles fournissent des informations utiles à la mise en œuvre des politiques sanitaires efficaces. C'est à ces enquêtes, si importantes dans la lutte contre les épidémies, que se sont intéressés les travaux de cette thèse.

Dans l'objectif de contribuer, même faiblement, à l'émergence des pays en voie de développement, plus particulièrement les pays subsahariens, ainsi qu'à l'épanouissement des populations desdits pays, nous avons considéré le contexte parfois contraignant de ces pays. Nos premiers travaux furent ainsi orientés vers le choix de l'outil le plus adapté pour mener les enquêtes épidémiologiques dans ce contexte.

C'est dans cet objectif que nous avions mené une étude comparative de plusieurs suites logicielles mobiles de collecte de données (Bokonda, et al., 2020). Cette étude comparative a suivi une méthodologie en quatre étapes pour chacune des suites logicielles considérées. Ces quatre étapes sont les suivantes : *collecte et sélection des papiers, lecture des documents et extraction des données, installation des outils de la suite logicielle et exécution du workflow de collecte, enfin stockage et visualisation des données*.

Les deux premières étapes ont permis de comparer les suites logicielles en se basant sur la littérature. Il s'agissait de comparer, entre autres, la fréquence d'utilisation de ces suites dans les pays en voie de développement ainsi que leur utilisation dans le domaine médical voire dans le domaine épidémiologique. Il s'est avéré que la suite logicielle ODK est celle qui a le plus été utilisée dans le domaine médical et dans les pays en voie de développement principalement dans les enquêtes épidémiologiques. Nous avions considéré les critères suivants : *Gratuit et Open Source, Documentation en ligne, La réactivité de l'équipe support, La fréquence des*

mises à jour du logiciel et Communauté d'utilisateurs. De ces cinq critères, ODK s'est distingué des autres suites avec une note de 5/5, suivi de ODK-X 4/5, puis de KoboToolBox et PendragonForms qui ont tous les deux 2,5/5.

Les deux dernières étapes ont permis d'obtenir une comparaison empirique. Nous avons, nous-même installé et exécuté toutes les suites concernées par notre étude. Les critères de comparaison furent les suivants : *conception de formulaires mobiles simples et personnalisés, conception de formulaires mobiles complexes et personnalisés, interopérabilité avec d'autres technologies/outils etlangages de programmation*. Sur ces critères toutes les suites s'égalent avec une note de 3/4.

Notre choix final s'est porté sur ODK, principalement grâce à son côté gratuit et open source, ainsi que sa popularité dans les pays subsahariens où il a déjà été utilisé à plusieurs reprises dans le cadre de la lutte contre les épidémies.

Après avoir effectué ce choix, nous avons dressé un état de l'art détaillé sur ODK qui a fait l'objet de deux publications (BOKONDA, et al., 2019), (BOKONDA, et al., 2020). Dans ces travaux, nous avons, entre autres, introduit dans la littérature un modèle fonctionnel de ODK. Il s'agit d'une architecture qui expose et formalise tous les cas d'utilisation de la suite ODK. Ces travaux ont, par ailleurs, résumé les utilisations de ODK dans différents pays et domaines. Ils ont révélé que la médecine est le domaine où ODK a été le plus utilisé depuis sa conception et qu'ODK est une suite spécialement conçue pour répondre aux besoins des pays en voie de développement.

Ayant identifié et étudié la suite logicielle à recommander pour les enquêtes liées aux épidémies dans les pays en voie de développement, il nous fallait par la suite, nous orienter vers l'analyse prédictive. C'est ce que nous avons fait dans la suite de nos travaux de recherche.

L'Etat de l'art sur les méthodes de ML pour l'analyse prédictive (Bokonda, et al., 2020) nous a permis d'identifier la méthode Random Forests comme la méthode la plus appropriée pour les études de prédiction épidémiologique. En effet, après avoir identifié l'apprentissage supervisé comme type d'apprentissage le plus utilisé dans les études prédictives, nos travaux ont montré que 50% des articles étudiés utilisaient ce type d'apprentissage dans le domaine médical. Ensuite, toujours sur la base des articles ayant fait l'objet de cet état de l'art, nous avions constaté qu'il y avait cinq méthodes du type apprentissage supervisé qui sont fréquemment utilisées dans le domaine médical. Ces cinq méthodes sont : Random Forests (RF), Artificial

Neural Network (ANN), Decision Tree (DT), Support Vector Machine (SVM) et Logistic Regression (LR). RF s'est distingué des autres en occupant la première place avec 23% des cas d'utilisation en médecine suivi de ANN, DT et SVM avec 20% chacune ensuite LR 17 %.

Une deuxième étude nous a permis de confirmer ces résultats. En effet, dans cette étude (BOKONDA, et al., 2021), présentée dans le chapitre 4 de ce mémoire de thèse, nous avons considéré quatre épidémies : la peste porcine Africaine (PPA), la dengue, la grippe ainsi que le Norovirus de l'huître. Nous y avons constaté que les méthodes RF et ANN étaient les seules à avoir été utilisées pour les quatre épidémies. RF prendra alors le dessus sur ANN lors de l'analyse des métriques obtenues par chacune d'elles pour chaque épidémie. Considérant les meilleurs scores, RF obtient la meilleure précision pour deux épidémies sur quatre (PPA et la dengue) alors que ANN a la meilleure précision pour une seule (le norovirus de l'huitre). En moyenne, la précision de RF pour les quatre épidémies était de 93,94 % alors que celle de ANN était de 91,69 %. Ces résultats sont soutenus par la conclusion des travaux de Gilles R. Ducharme (Ducharme, 2018) qui après avoir comparé six méthodes (Bn, RLog, SVM, RF, KNN, ANN) parvient à la conclusion suivante : « Le classifieur qui ressort de cet exercice avec les meilleurs scores est RF et ses variantes ».

Ainsi, après avoir identifié RF en se basant sur plusieurs études et différentes méthodologies de comparaison, nous avions entrepris la principale contribution de cette thèse : l'enrichissement de RF pour la rendre plus performante dans la lutte contre la propagation des épidémies.

Pour ce faire, nous nous sommes basés sur l'étude de (Buvana & Muthumayil, 2021). Dans celle-ci, les auteurs considèrent cinq méthodes (GNB, LR, SVM, KNN, DT) qu'ils ont appliquées sur un dataset clinique de la COVID-19. Les cinq méthodes ont été comparées sur la base de quatre métriques (Accuracy, Precision, Recall et F1score).

Notons à ce niveau que dans la lutte contre les épidémies, les acteurs peuvent soit agir en amont par des mesures de prévention pour éviter l'apparition d'une épidémie, qui sévit dans une région voisine soit en aval pour empêcher la résurgence d'une épidémie qui vient de s'achever. Un autre niveau de la lutte est celui qui se passe pendant l'épidémie : lutter contre la propagation de l'épidémie. Prendre les mesures qui empêchent que l'épidémie se propage davantage et ainsi limiter le nombre de personnes atteintes. C'est à ce niveau que se positionne la principale contribution de cette thèse.

Dans le travail de recherche ci-haut cité de (Buvana & Muthumayil, 2021), les auteurs obtiennent pour résultat que la méthode DT est meilleure aux quatre autres méthodes pour toutes les métriques considérées. Dans le but de comparer nos résultats aux leurs, nous avions appliqué RF sur le même dataset et utilisé les mêmes métriques. Nous avions alors trouvé que RF était meilleur que DT pour toutes les métriques. C'est à la suite de ce résultat que nous avions entrepris la modification de RF.

Afin de modifier la méthode RF, nous avons considéré son algorithme de base qui est *Forest RI* (Breiman, 2001). Nous avons ainsi procédé à la modification de cet algorithme en y introduisant des étapes supplémentaires de traitement de données au travers d'un autre algorithme que nous avons nommé *Var Cust*. La nouvelle forme de l'algorithme *Forest RI* comportant *Var Cust* nous l'avons nommé *Forest EP* (Epidemiological Prediction) et la méthode RF qui l'implémente RF EP (Random Forest for Epidemiological Prediction). Les quatre étapes que nous avions ajoutées via *Var Cust* sont les suivantes :

- La sélection de variables significatives.
- La normalisation des données.
- La réduction de dimension.
- La sélection de nouvelles variables.

Après la construction de RF EP (la version enrichie de Random Forests), nous l'avions appliquée sur le même dataset précédemment mentionné (Buvana & Muthumayil, 2021), et évaluée avec les mêmes métriques. RF EP s'est montré meilleure que RF par défaut principalement en ce qui concerne la métrique Recall. Cette amélioration permet d'atteindre l'objectif de lutter contre la propagation de l'épidémie. Ce gain de performance sur la métrique Recall signifie que RF EP est en mesure de détecter plus de patients positifs que RF par défaut pour un dataset donné. Autrement dit, lorsqu'un hôpital ou un centre de santé utilise RF EP, il augmente sa capacité à détecter tous les patients positifs. En d'autres termes, moins de personnes porteuses de la maladie et donc susceptibles de la transmettre aux autres pourront passer par les mailles des tests.

Notre dernière contribution mais pas la moindre, a consisté à proposer le modèle d'une suite logicielle mobile comportant un module d'analyse prédictive épidémiologique. En effet, nous avons proposé l'architecture d'une telle suite que nous avons nommée MEPS. La suite MEPS est une fusion entre ODK et RF EP. Nous avons non seulement proposé une architecture

complète de celle-ci, ainsi qu'une explication des workflows de ODK retenus dans la suite MEPS, mais aussi une implémentation de ladite suite.

8.2. Perspectives

Bien que les travaux de recherche effectués dans le cadre de cette thèse, via un modèle de Machine Learning, participent à la lutte contre les épidémies et plus précisément à la diminution de la propagation de celles-ci, il reste encore quelques améliorations possibles. Nous envisageons les pistes d'amélioration suivantes :

- Amélioration de RF EP pour couvrir d'autres axes de la lutte contre les épidémies : Cette thèse s'est focalisée sur un axe de la lutte contre les épidémies, celui de limiter leur propagation. Toutefois, il existe d'autres axes dans cette lutte, dont la lutte contre la résurgence d'une épidémie dans une région ou l'apparition dans une autre. De futurs travaux de recherche, voire des thèses, peuvent partir de cette thèse pour proposer des versions améliorées des RF EP qui vont intervenir dans ces deux autres axes. Il s'agira dans ce cas, soit d'utiliser l'historique d'une épidémie dans une région pour prédire la période probable de sa réapparition (pour ce qui concerne la résurgence) ou pour prédire une possible apparition dans une région voisine. Il s'agira en ce moment-là d'une prédiction spatio-temporelle, axe que ne couvrent pas les travaux de cette thèse.

- Ajout d'autres méthodes de ML dans la suite logicielle MEPS : l'extension d'ODK que nous avons proposée dans cette thèse et que nous avons nommée MEPS utilise un module d'analyse prédictive avec une seule méthode de Machine Learning, RF EP. La perspective d'amélioration qui en découle est d'enrichir ce module d'analyse prédictive en y ajoutant d'autres méthodes de ML, préalablement testées. Ainsi, l'utilisateur pourra avoir le choix entre plusieurs méthodes et pourra choisir celle qui correspond le mieux à son cas d'utilisation. Les différentes méthodes devraient être accompagnées d'une description pour faciliter le choix des utilisateurs.

- Automatisation de la connexion entre ODK et RF EP : ODK est un ensemble de composants logiciels conçus et développés pour fonctionner de manière interconnectée et automatique, après quelques configurations. RF EP permet de prédire les cas de contamination à une épidémie de manière efficace. La suite logicielle que nous avons proposée (MEPS) et qui se veut être une extension d'ODK utilise ODK et RF EP. ODK

d'une part pour collecter les données via ODK Collect et les stocker dans le serveur ODK puis RF EP d'autre part pour analyser et prédire. Néanmoins, l'intervention de l'homme pour exporter les données venant du serveur d'ODK vers Jupyter Notebook (plateforme d'implémentation de RF EP) est toujours nécessaire. Une piste d'amélioration consiste à automatiser cette liaison de sorte qu'après la collecte de données avec ODK Collect, les données conservées dans le serveur soient automatiquement exportées sous format csv et envoyées à Jupyter Notebook.

Annexe : Liste des publications

1. Articles dans des revues à comité de lecture indexés dans des bases internationales

Loola Bokonda, P., Sidibe, M., Souissi, N., & Ouazzani-Touhami, K. Machine Learning Model for Predicting Epidemics. Computers, **2023**, vol. 12, no 3, p. 54.
Indexation: *Scopus, ESCI (Web of Science), Dblp, Inspec, EBSCO, DOAJ*
Qualité: *Q2*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. A Practical Analysis of Mobile Data Collection Apps. International Journal of Interactive Mobile Technologies, **2020**, vol. 14, no 13.
Indexation: *Scopus, Dblp, Inspec, EBSCO, DOAJ*
Qualité: *Q3*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. Open data kit: mobile data collection framework for developing countries. International Journal of Innovative Technology and Exploring Engineering (IJITEE), **2019**, vol. 8, no 12, p. 4749-4754.
Indexation: *Scopus*
Qualité: *Q4*

2. Communications dans des conférences à comité de lecture et proceedings indexés dans des bases internationales

Kaneho, A. E. A., Zrira, N., **Bokonda, P. L.**, & Ouazzani-Touhami, K. (**2022**, December). A Survey on Existing Chatbots for Pregnant Women's Healthcare. In 2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS) (pp. 1-6). IEEE.

Indexation: *Scopus, IEEE, Dblp*

DIOR, Masrané Reoukadji, **Loola Bokonda P.**, ALIHAMIDI, Imam, et al. Smart Medical Devices to Help Patients and Health Workers: A Survey. In: 2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS). IEEE, **2022**. p. 1-5.

Indexation : *Scopus, IEEE, Dblp*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. Which Machine Learning method for outbreaks predictions? In: **2021** IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2021. p. 825-828.

Indexation: *Scopus, IEEE*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. Predictive analysis using machine learning: Review of trends and methods. In: **2020** International Symposium on Advanced Electrical and Communication Technologies (ISAECT). IEEE, **2020**. p. 1-6.

Indexation: *Scopus, IEEE*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. LISUNGIcovid19: Prototype of mobile application to help manage the way out of covid-19 crisis. In: Colloq. Inform. Sci. Technol., CIST. **2020**. p. 63-68.

Indexation: *Scopus, IEEE, Dblp*

Loola Bokonda, P., Ouazzani-Touhami, K., & Souissi, N. Mobile Data Collection Using Open Data Kit. In: Innovation in Information Systems and Technologies to Support Learning Research: Proceedings of EMENA-ISTL 2019 3. Springer International Publishing, **2020**. p. 543-550.

Indexation: *Springer*

Bibliographie

Anokwa, Y. et al. 2009. Open Source Data Collection in the Developing World. *IEEE Computer,* 10.42(10).

Baudon, D. 2000. Les paludismes en Afrique subsaharienne. *Afrique contemporaine,* Volume 195, pp. 36-45.

Bell, A. R., Ward, P. S., Killilea, M. E. & Tamal, M. E. H. 2016. Real-Time Social Data Collection in Rural Bangladesh via a 'Microtasks for Micropayments' Platform on Android Smartphones. *PLoS ONE,* Novembre.

Benedictow, O. J. 2004. The Black Death, 1346-1353: the complete history. *Boydell & Brewer*.

Bokonda, L. P., Ouazzani-Touhami, K. & Souissi, N. 2019. Open Data Kit: Mobile Data Collection Framework For Developing Countries.. *International Journal of Innovative Technology and Exploring Engineering,* 8 12, 8(12), pp. 4749-4754.

Bokonda, L. P., Ouazzani-Touhami, K. & Souissi, N. 2020. A Practical Analysis of Mobile Data Collection Apps. *International Journal of Interactive Mobile Technologies,* 14(13).

Bokonda, L. P., Ouazzani-Touhami, K. & Souissi, N. 2020. *LISUNGIcovid19: Prototype of mobile application to help manage the way out of covid-19 crisis*. Agadir, IEEE.

Bokonda, L. P., Ouazzani-Touhami, K. & Souissi, N. 2020. *Mobile data collection using Open Data Kit*.. Marrakech, Springer International Publishing.

Bokonda, L. P., Ouazzani-Touhami, K. & Souissi, N. 2020. *Predictive analysis using machine learning: Review of trends and methods*.. s.l., IEEE, pp. 1-6.

Bokonda, L. P., Ouzzani-Touhami, K. & Souissi, N. 2021. *Which Machine Learning method for outbreaks predictions?*. s.l., IEEE, pp. 825-828.

Bokonda, L. P., Sidibe, M., Souissi, N. & Ouazzani-Touhami, K. 2023. Machine Learning Model for Predicting Epidemics. *Computers,* 12(3), p. 54.

Brunette, W. 2017. *Extended Abstract: Building Mobile Application Frameworks for Disconnected Data Management*. Niagara Falls, New York, USA, ACM, pp. 15-16.

Brunette, W. et al. 2013. *Open Data Kit 2.0: Expanding and Refining Information Services for Developing Regions*. Jekyll Island, Georgia, ACM, p. Article No. 10.

Brunette, W. et al. 2017. *Open Data Kit 2.0: A Services-Based Application Framework for Disconnected Data Management.*. Niagara Falls, New York, USA, ACM, pp. 440-452.

Brunette, W. et al. 2013. *ODK Tables: Building Easily Customizable Information Applications on Android Devices*. Bangalore, India, s.n.

Brunette, W., Sundt, M., Ginsburg, A. & Borriello, G. 2013. *Customizing and improving medical workflows using ODK survey*. Cape Town, South Africa, ACM, p. Article No. 22.

Brunette, W. et al. 2015. *Optimizing Mobile Application Communication for Challenged Network Environments*. London, United Kingdom, ACM, pp. 167-175.

Buvana, M. & Muthumayil, K., 2021. Prediction of COVID-19 patient using supervised machine learning algorithm. *Sains Malaysiana,* pp. 2479-2497.

C. Hartung, A. L. Y. A. C. T. & W. Brunette, G. B. 2010. *Open data kit: tools to build information services for developing regions*. London, United Kingdom, ACM, p. Article No. 18.

Chaudhri, R., Brunette, W., Hemingway, B. & Borriello, G. 2013. *ODK sensors: an application-level sensor framework for Android devices*. Bangalore, India, s.n., p. Article No. 30.

Djomassi, L. D., Gessner, B. D., Andze, G. O. & Mballa, G. E. 2013. National surveillance data on the epidemiology of cholera in Cameroon. *The Journal of infectious diseases,* 208((suppl_1)), pp. S92-S97.

Fornace, K. M. et al. 2018. Use of mobile technology-based participatory mapping approaches to geolocate health facility attendees for disease surveillance in low resource settings . *International Journal of Health Geographics,* Juin.pp. 1-10.

Gajawada, S. K. 2019. *Chi-Square Test for Feature Selection in Machine learning.* [En ligne] Available at: https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223
[Accès le 08 03 2023].

Giduthuri, J. G. et al. 2014. Developing and validating a tablet version of an illness explanatory model interview for a public health survey in Pune, India. *PLOS ONE,* 18 Septembre.

Ginsburg, A. S. et al. 2015. mPneumonia: Development of an Innovative mHealth Application for Diagnosing and Treating Childhood Pneumonia and Other Childhood Illnesses in Low-Resource Settings. *PLoS ONE,* 15 Octobre.10(10).

Gottfried & Robert , S., 2010. Black death. *Simon and Schuster*.

Gupta, A. et al. 2013. *Simplifying and improving mobile based data collection*. Cape Town, South Africa, ACM, pp. 45-48.

Hong Hilary, Y., Worden, K. & Borriellor, G. 2011. *ODK Tables: Data Organization and Information Services on a Smartphone*. Bethesda, Maryland, USA, ACM, pp. 33-38.

Intermountain, 2020. *What's the difference between a pandemic, an epidemic, endemic, and an outbreak?*. [En ligne] Available at: https://intermountainhealthcare.org/blogs/topics/live-well/2020/04/whats-the-difference-between-a-pandemic-an-epidemic-endemic-and-an-outbreak/#:~:text=Let's%20start%20with%20basic%20definitions,a%20particular%20people%20or%20country [Accès le 17 03 2023].

Jain, R., Bhatt, H., Jeevanand, N. & Kumar, P. 2018. Mapping, Visualization, and Digitization of the Geo-Referenced Information: A case study on Road Network Development in Near Real Time . *International Research Journal of Engineering and Technology (IRJET)*, septembre.

Jeffers, V. F. et al. 2019. Trialling the use of smartphones as a tool to address gaps in small-scale fisheries catch data in southwest Madagascar. Janvier, Volume 99, pp. 267-274.

JeffreyCoker, F., Basinger, M. & Modi, V. 2010. [En ligne] Available at: https://s3.amazonaws.com/academia.edu.documents/30934121/Open-Data-Kit-Review-Article.pdf?response-content-disposition=inline%3B%20filename%3DOpen_Data_Kit_Implications_for_the_Use_o.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL

Jézégou , F. 2001. *DICOCITATIONS.* [En ligne] Available at: https://www.dicocitations.com/citations/citation-52270.php [Accès le 17 03 2023].

Jupyter, T. 2023. *Jupyter Notebook Official Web Site.* [En ligne] Available at: https://jupyter.org/ [Accès le 08 03 2023].

KBest in Scikit, 2007. *sklearn feature_selection SelectKBest.* [En ligne] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html [Accès le 08 03 2023].

Kipf, A. et al. 2016. A proposed integrated data collection, analysis and sharing platform for impact evaluation. *ELSEVIER.*

KoBoToolbox, T. 2021. *KoBoToolbox. Official Website.* [En ligne] Available at: https://www.kobotoolbox.org/ [Accès le 19 02 2023].

Koike, F. & Morimoto, N. 2018. Supervised forecasting of the range expansion of novel non-indigenous organisms: Alien pest organisms and the 2009 H1N1 flu pandemic. *Global Ecology and Biogeography,*, pp. 991-1000.

Lafricamobile, 2022. *Utilisation Des Smartphones En Afrique.* [En ligne] Available at: https://lafricamobile.com/fr/utilisation-des-smartphones-en-afrique/ [Accès le 17 03 2023].

LeMondeAfrique, 2020. *Paludisme, rougeole ou choléra : des épidémies plus meurtrières que le Covid-19 sévissent aujourd'hui en Afrique.* [En ligne] Available at: https://www.lemonde.fr/afrique/article/2020/05/20/en-afrique-subsaharienne-d-autres-epidemies-plus-meurtrieres-que-le-covid-19-continuent-de-sevir_6040293_3212.html [Accès le 17 03 2023].

Little, A. et al. 2013. Meeting Community Health Worker Needs for Maternal Health Care Service Delivery Using Appropriate Mobile Technologies in Ethiopia. *PLoS ONE,* 29 Octobre.8(10).

Macharia, P. et al. 2013. Open data kit, a solution implementing a mobile health information system to enhance data management in public health. *IEEE.*

Medhanyie, A. A. et al. 2015. Health workers' experiences, barriers, preferences and motivating factors in using mHealth forms in Ethiopia . *Human resources for health,* 15 Janvier, 13(1), pp. 1-10.

Monde, L. 2020. *Paludisme, rougeole ou choléra : des épidémies plus meurtrières que le Covid-19 sévissent aujourd'hui en Afrique.* [En ligne] Available at: https://www.lemonde.fr/afrique/article/2020/05/20/en-afrique-subsaharienne-d-autres-epidemies-plus-meurtrieres-que-le-covid-19-continuent-de-sevir_6040293_3212.html
[Accès le 17 03 2023].

Morvan, J. 2022. *Epidémie d'infections à virus Chikungunya en Ethiopie.* [En ligne] Available at: https://www.mesvaccins.net/web/news/18950-epidemie-d-infections-a-virus-chikungunya-en-ethiopie
[Accès le 17 03 2023].

Mosavi, A., Ozturk, P. & Chau, K. W. 2018. Flood prediction using machine learning models: Literature review.. *Water,* 10(11), p. 1536.

ODK Central, 2023. [En ligne] Available at: https://docs.getodk.org/central-intro/ [Accès le 08 03 2023].

ODK, T. 2023. *Open Data Kit. Official Website.* [En ligne] Available at: https://docs.getodk.org/ [Accès le 19 02 2023].

ODK-X, T. 2023. *ODK-X documentation.* [En ligne] Available at: https://docs.odk-x.org/ [Accès le 19 02 2023].

OMS, 2018. *Maladie à virus Ebola.* [En ligne] Available at: https://apps.who.int/mediacentre/factsheets/fs103/fr/index.html [Accès le 17 03 2023].

OMS, 2020. *COVID-19 – Chronologie de l'action de l'OMS.* [En ligne] Available at: https://www.who.int/fr/news/item/27-04-2020-who-timeline---covid-19
[Accès le 01 05 2023].

Pandas, T. 2023. *Pandas Official Web Site.* [En ligne] Available at: https://pandas.pydata.org/docs/ [Accès le 08 03 2023].

Patterson, K. B. & Runge, T. 2002. Smallpox and the native American. *The American journal of the medical sciences,* 323(4), pp. 216-222.

PendragonForms, T. 2021. *PendragonForms. Official Website.* [En ligne] Available at: http://www.pendragonforms.com/ [Accès le 19 02 2023].

Prabhu , M. & Gergen, J. 2022. *VaccinesWork - Les sept pandémies les plus meurtrières de l'histoire.* [En ligne] Available at: https://www.gavi.org/fr/vaccineswork/sept-pandemies-meurtrieres-histoire [Accès le 17 03 2023].

RFI, 2022. *Abidjan touchée par une cinquième épidémie de dengue due à l'insalubrité.* [En ligne] Available at: https://www.rfi.fr/fr/afrique/20221029-abidjan-touch%C3%A9e-par-une-cinqui%C3%A8me-%C3%A9pid%C3%A9mie-de-dengue-due-%C3%A0-l-insalubrit%C3%A9 [Accès le 17 03 2023].

Sabbatani, S., Manfredi, R. & Fiorino, S. 2012. The Justinian plague (part one). *Le infezioni in medicina,* 20(2), pp. 125-139.

Scikit-learn, T. 2023. *Scikit-learn official site.* [En ligne] Available at: https://scikit-learn.org/stable/ [Accès le 08 03 2023].

SciPy, 2023. *SciPy official website.* [En ligne] Available at: https://scipy.org/ [Accès le 08 03 2023].

Soh, J., Singh, P., Soh, J. & Singh, P. 2020. Machine learning operations. *Data Science Solutions on Azure: Tools and Techniques Using Databricks and MLOps,* pp. 259-279.

Tapak, L., Hamidi, O., Fathian, M. & Karami, M. 2019. Comparative evaluation of time series models for predicting influenza outbreaks:Application of influenza-like illness data from sentinel sites of healthcare centers in Iran. *BMC research notes,* p. 353.

Taubenberger, J. K. & Morens, D. M. , 2006. 1918 Influenza: the mother of all pandemics. *Revista Biomedica,* 17(1), pp. 69-79.

Van Ree, E. 2003. *The Political Thought of Joseph Stalin: A study in twentieth century revolutionary patriotism.* s.l.:Routledge.

Cayla B. 2021. Biais & Variance … dilemme ou compromis ?. [En ligne] Available at: https://datacorner.fr/biais-variance/ [Accès le 16 06 2023].

Kumara B., Anantha Padmanabhan S., "Optimized Data Collection and Computation using Internet of Things (IoT)." (2019). International Journal of Innovative Technology and Exploring Engineering, VOLUME-8 ISSUE-10, AUGUST 2019, REGULAR ISSUE, 8(10), 870–873. https://doi.org/10.35940/ijitee.j9047.0881019.

Ryong Lee, Minwoo Park, Sang-HwanLee, "An Advanced IoT Data Collection Service for Data-centric Smart Cities." (2019). International Journal of Innovative Technology and Exploring Engineering, Volume-8 Issue-8S2, June 2019.

El Arass M., Souissi N.: Data Lifecycle: From Big Data to Smart Data. In: IEEE 5th International Congress on Information Science and Technology (CiSt), pp. 80-87, Marrakech (2018).

Tikito I., Souissi N.: Data Collect Requirements Model. In: Proceeding of the BDCA2017, 28-30 March, Tetuan (2017).

El Arass M., Tikito I., Souissi N.: Data lifecycle analysis: towards intelligent cycle. In: Proceeding of The second International Conference on Intelligent Systems and Computer Vision, ISCV2017, Fs 17-19 April, Fez (2017).

CyberTracker, August 2019. http://cybertracker.co.za.

Pendragon Forms, August 2019. http://pendragonsoftware.com.

FrontlineSMS, August 2019. http://frontlinesms.com.

EpiSurveyor, August 2019. http://datadyne.org.

CommCare, August 2019. http://dimagi.com/commcare.

JavaRosa, August 2019. http://bitbucket.org/javarosa.

Anokwa, Y., Hartung, C., Brunette, W., Borriello, G. and Lerer, A. 2009. Open Source Data Collection in the Developing World. Computer, vol. 42, no. 10, pp. 97–99 (Oct. 2009). DOI:https://doi.org/10.1109/mc.2009.328.

Open data kit history, August 2019. https://opendatakit.org/community/history/.

Jeffers, V.F., Humber, F., Nohasiarivelo, T., Botosoamananto, R., Anderson, L.G., 2019. Trialling the use of smartphones as a tool to address gaps in small-scale fisheries catch data in southwest Madagascar. Marine Policy 99, 267–274. https://doi.org/10.1016/j.marpol.2018.10.040.

Hartung, C., Lerer, A., Anokwa, Y., Tseng, C., Brunette, W. and Borriello, G. 2010. Open Data Kit: Tools to Build Information Services for Developing Regions. Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development - ICTD ’10 (2010).

Brunette, W., Sundt, M., Dell, N., Chaudhri, R., Breit, N. and Borriello, G. 2013. Open Data Kit 2.0: Expanding and Refining Information Services for Developing Regions. Proceedings of the 14th Workshop on Mobile Computing Systems and Applications - HotMobile ’13 (2013).

Brunette, W., Sudar, S., Sundt, M., Larson, C., Beorse, J. and Anderson, R. 2017. Open Data Kit 2.0: A Services-Based Application Framework for Disconnected Data Management. Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys ’17 (2017).

Brunette, W. 2017. Building Mobile Application Frameworks for Disconnected Data Management. Proceedings of the 2017 Workshop on MobiSys 2017 Ph.D. Forum - Ph.D. Forum ’17 (2017).

Hong, Y., Worden, H.K. and Borriello, G. 2011. ODK Tables: Data Organization and Information Services on a Smartphone. Proceedings of the 5th ACM workshop on Networked systems for developing regions - NSDR ’11 (2011).

Chaudhri, R., Brunette, W., Goel, M., Sodt, R., VanOrden, J., Falcone, M. and Borriello, G. 2012. Open Data Kit Sensors: Mobile Data Collection with Wired and Wireless Sensors. Proceedings of the 2nd ACM Symposium on Computing for Development - ACM DEV ’12 (2012).

Brunette, W., Sodt, R., Chaudhri, R., Goel, M., Falcone, M., Van Orden, J. and Borriello, G. 2012. Open Data Kit Sensors: A Sensor Integration Framework for Android at the Application-Level. Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12 (2012).

Brunette, W., Sundt, M., Ginsburg, A. and Borriello, G. 2013. Customizing and improving medical workflows using ODK survey. Proceedings of the 4th Annual Symposium on Computing for Development - ACM DEV-4 '13 (2013).

Brunette, W., Vigil, M., Pervaiz, F., Levari, S., Borriello, G. and Anderson, R. 2015. Optimizing Mobile Application Communication for Challenged Network Environments. Proceedings of the 2015 Annual Symposium on Computing for Development - DEV '15 (2015).

Chaudhri, R., Brunette, W., Hemingway, B. and Borriello, G. 2013. ODK Sensors: an Application-level Sensor Framework for Android devices. Proceedings of the 3rd ACM Symposium on Computing for Development - ACM DEV '13 (2013).

Brunette, W., Sudar, S., Worden, N., Price, D., Anderson, R. and Borriello, G. 2013. ODK Tables: Building Easily Customizable Information Applications on Android Devices. Proceedings of the 3rd ACM Symposium on Computing for Development - ACM DEV '13 (2013).

Gupta, A., Thapar, J., Singh, A., Singh, P., Srinivasan, V. and Vardhan, V. 2013. Simplifying and improving mobile based data collection. Proceedings of the Sixth International Conference on Information and Communications Technologies and Development Notes - ICTD '13 - volume 2 (2013).

Kipf, A., Brunette, W., Kellerstrass, J., Podolsky, M., Rosa, J., Sundt, M., Wilson, D., Borriello, G., Brewer, E., Thomas, E., 2016. A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering 1, 36–44. https://doi.org/10.1016/j.deveng.2015.12.002.

Ronak Jain, Himansi Bhatt, NithyaJeevanand, "Mapping, Visualization, and Digitization of the Geo-Referenced Information: A case study on Road Network Development in Near Real Time."(2018). International Research Journal of Engineering and Technology (IRJET).Volume: 05 Issue: 09, Sept. 2018.

Macharia P., Muluve E., Lizcano J., Cleland C., Cherutich P., & Kurth A. (Januray 2013). Open data kit, a solution implementing a mobile health information system to enhance data management in public health. IEEE.

Signore, A., 2016. Mapping and sharing agro-biodiversity using Open Data Kit and Google Fusion Tables. Computers and Electronics in Agriculture 127, 87–91. https://doi.org/10.1016/j.compag.2016.06.006.

Jeffrey-Coker F., Basinger M., and Modi V. Open Data Kit: Implications for the Use of Smartphone Software Technology for Questionnaire Studies in International Development, March 2010. http://modi.mech.columbia.edu/2010/04/open-datakit.

Giduthuri JG, Maire N, Joseph S, Kudale A, Schaetti C, Sundaram N, et al. Developing and Validating a Tablet Version of an Illness Explanatory Model Interview for a Public Health Survey in Pune, India. Chuang J-H, editor. PLoS ONE. Public Library of Science (PLoS); 2014; 9: e107374. doi:10.1371/journal.pone.0107374.

Tom-Aba D, Olaleye A, Olayinka AT, Nguku P, Waziri N, Adewuyi P, et al. Innovative Technological Approach to Ebola Virus Disease Outbreak Response in Nigeria Using the Open Data Kit and Form Hub Technology. Harper DM, editor. PLOS ONE. Public Library of Science (PLoS); 2015; 10: e0131000. doi:10.1371/journal.pone.0131000.

Bell AR, Ward PS, Killilea ME, Tamal MEH. Real-Time Social Data Collection in Rural Bangladesh via a "Microtasks for Micropayments" Platform on Android Smartphones. Paul R, editor. PLOS ONE. Public Library of Science (PLoS); 2016; 11: e0165924. doi:10.1371/journal.pone.0165924

Rajput ZA, Mbugua S, Amadi D et al. Evaluation of an Android-based mHealth system for population surveillance in developing countries. Journal of the American Medical Informatics Association 2012;19:655–9.

Medhanyie AA, Little A, Yebyo H, et al. Health workers' experiences, barriers, preferences and motivating factors in using mHealth forms in Ethiopia. Hum ResourHealth 2015; 13(2): 1–22.

Tridane, A., Raja, A., Gaffar, A., Lindquist, T., &Prabadi, K. (2014). Android and ODK based data collection framework to aid in epidemiological analysis. Online Journal of Public Health Informatics, 5(3). https://doi.org/10.5210/ojphi.v5i3.4996

Fornace, K.M.; Surendra, H.; Abidin, T.R.; Reyes, R.; Macalinao, M.L.; Stresman, G.; Luchavez, J.; Ahmad, R.A.; Supargiyono, S.; Espino, F.; et al. Use of mobile technology-based participatory mapping approaches to geolocate health facility attendees for disease surveillance in low resource settings. Int. J. HealthGeogr. 2018, 17, 21.

Pavluck A, Chu B, Mann Flueckiger R, Ottesen E. Electronic Data Capture Tools for Global Health Programs: Evolution of LINKS, an Android-, Web-Based System. Brooker S, editor. PLoSNeglected Tropical Diseases. Public Library of Science (PLoS); 2014;8: e2654. doi:10.1371/journal.pntd.0002654.

Shiferaw, S., Workneh, A., Yirgu, R., Dinant, G.-J., &Spigt, M. (2018). Designing mHealth for maternity services in primary health facilities in a low-income setting – lessons from a partially successful implementation. BMC Medical Informatics and Decision Making, 18(1). https://doi.org/10.1186/s12911-018-0704-9.

Little, A., Medhanyie, A., Yebyo, H., Spigt, M., Dinant, G.-J., & Blanco, R. (2013). Meeting Community Health Worker Needs for MaternalHealth Care Service Delivery Using Appropriate Mobile Technologies in Ethiopia. PLoS ONE, 8(10), e77563. https://doi.org/10.1371/journal.pone.0077563.

Ginsburg AS, Delarosa J, Brunette W, Levari S, Sundt M, Larson C, et al. mPneumonia: Development of an Innovative mHealth Application for Diagnosing and Treating Childhood Pneumonia and Other Childhood Illnesses in Low-Resource Settings. Jordan IK, editor. PLOS ONE. Public Library of Science (PLoS); 2015;10: e0139625. doi:10.1371/journal.pone.0139625.

Google play store permission, August 2009. https://play.google.com/about/privacy-security-deception/permissions/

Montreal University: http://www.iro.umontreal.ca/ nie/IFT3335/Introduction.html, accessedon 13 june 2020.

CNRS LE JOURNAL, Yaroslav Pigenet : https://lejournal.cnrs.fr/articles/des-machines-enfin-intelligentes.

Zinebi, K., Souissi, N., Tikito, K. (2018). Driver Behavior Analysis Methods: Applications oriented study. In Proceedings of the 3rd International Conference on Big Data, Cloud and Application (BDCA 2018), April 2018 in Kenitra Morocco.

Lima, M. S. M., & Delen, D. (2020). Predicting and explaining corruption across countries: A machine learning approach. Government Information Quarterly, 37(1), 101407.

Kaur, H., & Kumari, V. (2018). Predictive modelling and analytics for diabetes using a machine learning approach. Applied computing and informatics.

Casta˜n´on, J. (10). Machine Learning Methods that Every Data Scientist Should Know. Consultado em Outubro, 16, 2019.

Lanier, P., Rodriguez, M., Verbiest, S., Bryant, K., Guan, T., & Zolotor, A. (2020). Preventing infant maltreatment with predictive analytics: applying ethical principles to evidence-based child welfare policy. Journal of family violence, 35(1), 1-13.

Patel, N. J., & Jhaveri, R. H. (2015, January). Detecting packet dropping nodes using machine learning techniques in Mobile ad-hoc network: A survey. In 2015 International Conference on Signal Processing and Communication Engineering Systems (pp. 468-472). IEEE.

Moujahid, A., Tantaoui, M. E., Hina, M. D., Soukane, A., Ortalda, A., ElKhadimi, A., & Ramdane-Cherif, A. (2018, June). Machine learning techniques in ADAS: A review. In 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE) (pp. 235-242). IEEE.

Yang, H., Xie, X., & Kadoch, M. (2020). Machine Learning Techniques and A Case Study for Intelligent Wireless Networks. IEEE Network, 34(3), 208-215.

Johnston, S. S., Morton, J. M., Kalsekar, I., Ammann, E. M., Hsiao, C. W., & Reps, J. (2019). Using machine learning applied to real-world healthcare data for predictive analytics: an applied example in bariatric surgery. Value in Health, 22(5), 580-586.

Lorenzo, A. J., Rickard, M., Braga, L. H., Guo, Y., & Oliveria, J. P. (2019). Predictive Analytics and Modeling Employing Machine Learning Technology: The Next Step in Data

Sharing, Analysis, and Individualized Counseling Explored With a Large, Prospective Prenatal Hydronephrosis Database. Urology, 123, 204-209.

Model-based Machine Learning: http://www.mbmlbook.com/Introduction.html

Rezaianzadeh, A., Dastoorpoor, M., Sanaei, M., Salehnasab, C., Mohammadi, M. J., & Mousavizadeh, A. (2020). Predictors of length of stay in the coronary care unit in patient with acute coronary syndrome based on data mining methods. Clinical Epidemiology and Global Health, 8(2), 383-388.

Bergsten, T. M., Nicholson, A., Donnino, R., Wang, B., Fang, Y., & Natarajan, S. (2020). Predicting adults likely to develop heart failure using readily available clinical information: An analysis of heart failure incidence using the NHEFS. Preventive Medicine, 130, 105878.

Xu, L., Wu, J., Lu, W., Yang, C., & Liu, H. (2019, December). Application of the Albumin-Bilirubin Grade in Predicting the Prognosis of Patients With Hepatocellular Carcinoma: A Systematic Review and Meta-Analysis. In Transplantation Proceedings (Vol. 51, No. 10, pp. 3338-3346). Elsevier.

Robert, B. M., Brindha, G. R., Santhi, B., Kanimozhi, G., & Prasad, N. R. (2019). Computational models for predicting anticancer drug efficacy: A multi linear regression analysis based on molecular, cellular and clinical data of oral squamous cell carcinoma cohort. Computer methods and programs in biomedicine, 178, 105-112.

Ji, G. W., Zhu, F. P., Xu, Q., Wang, K., Wu, M. Y., Tang, W. W., ... & Wang, X. H. (2019). Machine-learning analysis of contrast-enhanced CT radiomics predicts recurrence of hepatocellular carcinoma after resection: A multi-institutional study. EBioMedicine, 50, 156-165.

Lai, H., Huang, H., Keshavjee, K., Guergachi, A., & Gao, X. (2019). Predictive models for diabetes mellitus using machine learning techniques. BMC endocrine disorders, 19(1), 1-9.

Lee, Y., Ragguett, R. M., Mansur, R. B., Boutilier, J. J., Rosenblat, J. D., Trevizol, A., ... & Chan, T. C. (2018). Applications of machine learning algorithms to predict

therapeutic outcomes in depression: a meta-analysis and systematic review. Journal of affective disorders, 241, 519-532.

Bumberger, A., Clauser, P., Kolta, M., Kapetas, P., Bernathova, M., Helbich, T. H., ... & Baltzer, P. A. (2019). Can we predict lesion detection rates in second-look ultrasound of MRI-detected breast lesions? A systematic analysis. European journal of radiology, 113, 96-100.

Araz, O. M., Olson, D., & Ramirez-Nafarrate, A. (2019). Predictive

analytics for hospital admissions from the emergency department using

triage information. International Journal of Production Economics, 208,

199-207.

Huang, L., Shea, A. L., Qian, H., Masurkar, A., Deng, H., & Liu, D. (2019). Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. Journal of biomedical informatics, 99, 103291.

Hill, B. L., Brown, R., Gabel, E., Rakocz, N., Lee, C., Cannesson, M., ... & Maoz, U. (2019). An automated machine learning-based model predicts postoperative mortality using readily-extractable preoperative electronic health record data. British Journal of Anaesthesia, 123(6), 877-886.

Ramesh, D., & Katheria, Y. S. (2019). Ensemble method based predictive model for analyzing disease datasets: a predictive analysis approach. Health and Technology, 9(4), 533-545.

Han, D. H., Lee, S., & Seo, D. C. (2020). Using machine learning to predict opioid misuse among US adolescents. Preventive medicine, 130, 105886.

De Castro Vieira, J. R., Barboza, F., Sobreiro, V. A., & Kimura, H. (2019). Machine learning models for credit analysis improvements: Predicting low-income families' default. Applied Soft Computing, 83, 105640.

Nosseir, A., & Fathy, Y. (2020). A Mobile Application for Early Prediction of Student Performance Using Fuzzy Logic and Artificial Neural Networks.

Choi, J., Gu, B., Chin, S., & Lee, J. S. (2020). Machine learning predictive model based on national data for fatal accidents of construction workers. Automation in Construction, 110, 102974.

Sam, E. F., Brijs, K., Daniels, S., Brijs, T., &Wets, G. (2020). Testing the convergent-and predictive validity of a multi-dimensional belief-based scale for attitude towards personal safety on public bus/minibus for longdistance trips in Ghana: A SEM analysis. Transport policy, 85, 67-79.

Maione, C., Barbosa Jr, F., & Barbosa, R. M. (2019). Predicting the botanical and geographical origin of honey with multivariate data analysis and machine learning techniques: A review. Computers and Electronics in Agriculture, 157, 436-446.

Talaei-Khoei, A., Tavana, M., & Wilson, J. M. (2019). A predictive analytics framework for identifying patients at risk of developing multiple medical complications caused by chronic diseases. Artificial Intelligence in Medicine, 101, 101750.

Basnet, S. M., Aburub, H., & Jewell, W. (2019). Residential demand response program: Predictive analytics, virtual storage model and its optimization. Journal of Energy Storage, 23, 183-194.

Gray, C. C., & Perkins, D. (2019). Utilizing early engagement and machine learning to predict student outcomes. Computers & Education, 131, 22-32.

Ge, F., Li, Y., Yuan, M., Zhang, J., & Zhang, W. (2020). Identifying predictors of probable posttraumatic stress disorder in children and adolescents with earthquake exposure: a longitudinal study using a machine learning approach. Journal of affective disorders, 264, 483-493.

Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., & Van Erven, G. (2019). Educational data mining: Predictive analysis of academic performance of public-school students in the capital of Brazil. Journal of Business Research, 94, 335-343.

Zolbanin, H. M., Delen, D., Crosby, D., & Wright, D. (2019). A Predictive Analytics-Based Decision Support System for Drug Courts. Information Systems Frontiers, 1-20.

Zolbanin, H. M., & Delen, D. (2018). Processing electronic medical records to improve predictive analytics outcomes for hospital readmissions. Decision Support Systems, 112, 98-110.

LEFIGARO: https://www.lefigaro.fr/sciences/dossier/virus-chinecoronavirus-pneumonie, accessed on 20 june, 2020.

The First Few X (FFX) Cases and contact investigation protocol for 2019-novel coronavirus (2019-nCoV) infection, version 2 : https://www.who.int/publications/i/item/the-first-few-x-(ffx)-cases-andcontact-investigation-protocol-for-2019-novel-coronavirus-(2019-ncov)-infection, accessed on 01 june, 2020.

Downloadable minimum reporting form https: //forum.getodk.org/t/whocovid-19-minimum-reporting-form/25647, accessed on 01 june, 2020.

StopCovid: https://www.economie.gouv.fr/appli-stop-covid-disponible, accessed on 05 june, 2020.

Wiqaytna: https://www.wiqaytna.ma/, accessed on 10 jun, 2020.

Tikito, I., and Souissi, N. (2020). Towards a systematic collect data process. International Journal of Big Data Intelligence, 1(1), 1. https://doi.org/10.1504/ijbdi.2020.10026663

Tikito, I., and Souissi, N. (2020). Smart Data Collection in Mobile Edge Computing Environment. In Proceedings of the 4th International Conference on Intelligent Systems and Computer Vision (IEEE ISCV 2020), Fez, Morocco, June 09-11, 2020.

Improving mapping for Ebola response through mobilising a local community with self-owned smartphones: Tonkolili District, Sierra Leone, January 2015 Nic Lochlainn L.M., Gayton I., Theocharopoulos G., Edwards R., Danis K., Kremer R., Kleijer K., (...), Caleo G. (2018) PLoS ONE, 13 (1) , art. no. e0189959

Tom-Aba et al.(2015). Innovative Technological Approach to Ebola Virus Disease Outbreak Response in Nigeria Using the Open Data Kit and Form Hub Technology. PLOS ONE, 10(6), e0131000. https://doi.org/10.1371/journal.pone.0131000

ODK in Liberia: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6675929/, accessed on 13 june, 2020.

ODK in DRC: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6675929/, accessed on 13 jun, 2020.

Tikito, I., El Arass, M., and Souissi, N. (2019). Meta-Analysis of Data Collect Methods. Journal of Computer Science, 15(8), 1184–1194. https://doi.org/10.3844/jcssp.2019.1184.1194

China App: https://edition.cnn.com/2020/04/15/asia/china-coronavirusqr-code-intl-hnk/index.html, accessed on 15 jun, 2020.

GH Covid19 Tracker: https://ghcovid19.com/, accessed on 15 june, 2020.

StopKorona: https://stop.koronavirus.gov.mk/en, accessed on 15 june, 2020.

eRouska: https://erouska.cz/, accessed on 15 june, 2020. [21] Smittestoppp: https://helsenorge.no/SiteCollectionDocuments/korona/smittestoppexplainer-engelsk-2020-04-24.pdf, accessed on 15 june, 2020.

TraceTogether: https://www.tracetogether.gov.sg/, accessed on 15 june, 2020.

Second phase of lockdown exit in Morocco: https://www.challenge.ma/deconfinement-le-maroc-passe-en-zone-1-a-lexception-de-tanger-larache-kenitra-et-marrakech-145531/, accessed on 25 june, 2020.

La peste porcine africaine : Questions – Réponses https://agriculture.gouv.fr/la-peste-porcine-africaine-questions-reponses, accessed on 10 december 2020.

Liang, R., Lu, Y., Qu, X., Su, Q., Li, C., Xia, S., ... & Niu, B. (2020). Prediction for global African swine fever outbreaks based on a combination of random forest algorithms and meteorological data. Transboundary and emerging diseases, 67(2), 935-946.

Iqbal, N., & Islam, M. (2019). Machine learning for dengue outbreak prediction: A performance evaluation of different prominent classifiers. Informatica, 43(3).

Anno, S., Hara, T., Kai, H., Lee, M. A., Chang, Y., Oyoshi, K., ... & Tadono, T. (2019). Spatiotemporal dengue fever hotspots associated with climatic factors in taiwan including outbreak predictions based on machine-learning. Geospatial health, 14(2).

Raja, D. B., Mallol, R., Ting, C. Y., Kamaludin, F., Ahmad, R., Ismail, S., ... & Sundram, B. M. (2019). Artificial intelligence model as predictor for dengue outbreaks. Malaysian Journal of Public Health Medicine, 19(2), 103-108.

Agarwal, N., Koti, S. R., Saran, S., & Senthil Kumar, A. (2018). Data mining techniques for predicting dengue outbreak in geospatial domain using weather parameters for New Delhi, India. Curr. Sci, 114(11), 2281-2291.

Muurlink, O. T., Stephenson, P., Islam, M. Z., & Taylor-Robinson, A. W. (2018). Long-term predictors of dengue outbreaks in Bangladesh: A data mining approach. Infectious Disease Modelling, 3, 322-330.

Bhatt, S., et al., The global distribution and burden of dengue. Nature, 2013. 496(7446): p. 504–507.

Dengue et dengue s´ev`ere https://www.who.int/fr/newsroom/fact-sheets/detail/dengue-and-severedengue#::text=L'incidence%20mondiale%20de%20la,sont%20g%C3%A9n%C3%A9ralement%20b%C3%A9nignes%20et%20asymptomatiques, accessed on 05 december 2020

Brady, O.J., et al., Refining the global spatial limits of dengue virus transmission by evidence-based consensus. PLOS Neglected Tropical Diseases, 2012. 6(8): p. e1760.

La grippe, une épidémie saisonnière https://www.santepubliquefrance.fr/maladies-et-traumatismes/maladieset-infections-respiratoires/grippe/articles/la-grippe-une-epidemiesaisonniere#::text=En , accessed on 03 december 2020

Lutter contre la grippe au Ghana https://www.who.int/bulletin/volumes/90/4/12-040412/fr/ , accessed on 20 november 2020

Tapak, L., Hamidi, O., Fathian, M., & Karami, M. (2019). Comparative evaluation of time series models for predicting influenza outbreaks: Application of influenza-like illness data from sentinel sites of healthcare centers in Iran. BMC research notes, 12(1), 353.

Grippe saisonni`ere https://www.who.int/fr/news-room/factsheets/detail/influenza-(seasonal) , accessed 18 november 2020

Koike, F., & Morimoto, N. (2018). Supervised forecasting of the range expansion of novel non-indigenous organisms: Alien pest organisms and the 2009 H1N1 flu pandemic. Global Ecology and Biogeography, 27(8), 991-1000.

Chenar, S. S., & Deng, Z. (2018). Development of genetic programmingbased model for predicting oyster norovirus outbreak risks. Water research, 128, 20-37.

Chenar, S. S., & Deng, Z. (2018). Development of artificial intelligence approach to forecasting oyster norovirus outbreaks along Gulf of Mexico coast. Environment international, 111, 212-223.

Thomas, A., Le Saux, J. C., Ollivier, J., Maalouf, H., Pommepuy, M., & Le Guyader, S. (2011). Norovirus et huˆıtres: de la terre `a la mer!. Virologie, 15(6), 353-360.

Wu, Y., Yang, Y., Nishiura, H., & Saitoh, M. (2018, June). Deep learning for epidemiological predictions. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 1085-1088).

Ducharme, G. R. (2018). Crit`eres de qualit´e d'un classifieur g´en´eraliste.

Fathima, A. S., & Manimeglai, D. (2015). Analysis of Significant Factors for Dengue Infection Prognosis Using the Random Forest Classifier. Analysis, 6(2). https://doi.org/10.14569/IJACSA.2015.060235

Fathima, A., & Manimegalai, D. (2012). Predictive analysis for the arbovirus-dengue using svm classification. International Journal of Engineering and Technology, 2(3), 521-7.

Tanner, L., Schreiber, M., Low, J. G., Ong, A., Tolfvenstam, T., Lai, Y. L., ... & Simmons, C. P. (2008). Decision tree algorithms predict the diagnosis and outcome of dengue fever in the early phase of illness. PLoS Negl Trop Dis,2(3), e196. https://doi.org/10.1371/journal.pntd.0000196

Brunette, W.: Building mobile application frameworks for disconnected data management. In: Proceedings of the 2017 Workshop on MobiSys 2017 Ph.D. Forum - Ph.D. Forum (2017)

Wei-Chih, L., Tierney, M., Chen, J., Kazi, F., Hubard, A., Pasquel, J.G., Subramanian, L., Rao, B.: UjU: SMS-based applications made easy. In: Proceedings of the First ACM Symposium on Computing for Development - ACM DEV (2010)