UMONS
Université de Mons

Faculté Polytechnique

DiabManager

Projet d'ingénierie informatique - Rapport final

Groupe 3
Mathis DELEHOUZÉE
Gustavo MAGAÑA LÓPEZ

Dirigé par Monsieur
Mohammed BENJELLOUN

Année académique 2018-2019

Remerciements

Nos remerciements vont à tous ceux qui ont participés de près ou de loin à l'organisation et la mise en place de notre projet. Nous tenons à remercier plus particulièrement Monsieur Benjelloun pour ses bons conseils et corrections durant les différents points de contrôle.

Mons, le 28 avril 2019

Notre équipe

FIGURE 1 – Mathis Delehouzée & Gustavo Magaña López

Gustavo Magaña López :
Étudiant erasmus en BA3 option informatique & gestion.
Université de Guanajuato, Méxique

Mathis Delehouzée :
Étudiant en BA3 option informatique & gestion.

Table des matières

Abréviations

SVR : Support vector regression

SVM : Supor vector machine

CGM : Continuous glucose monitoring

TG : Taux de glucose

T1D : Diabète de type 1

HP : Horizon de prédiction

ARIMA : Autoregressive moving average

RBF : Radial basis function

RMSE : Root-mean-square error

MAPE : Mean Absolute Percentage Error

RN : Réseau de neuronne

Introduction

Dans le cadre de notre cours d'ingénierie informatique, il nous a été demandé de réaliser un projet dans le domaine de l'IT. Malgré les nombreux projets intéressants proposés, nous avons décidé de réaliser un projet personnel portant sur le diabète de type 1.

Ce choix s'est fait tout naturellement car il s'agit d'un sujet dans lequel nous sommes déjà partiellement impliqué. C'est cette volonté de vouloir faire avancer la recherche dans ce domaine qui nous a pousser à entreprendre ce projet. Contribuer à l'amélioration de la qualité de vie de millions de personnes.

Bien entendu, nous n'avons pas la prétention de penser que nous pourrons faire un apport significatif dans la recherche sur ce domaine. Notre principal objectif est de nous familiariser avec l'utilisation du machine learning et son implémentation.

1 Contexte et motivation du projet

1.1 Le diabète

Le diabète de type 1 (T1D) est une maladie chronique touchant de plus en plus de personnes à travers le monde. En 2017 on comptait 425 millions de diabétique. Pour vous donner une représentation, cela représente 38 fois la population belge. Cette maladie pourrait, d'ici 2030, devenir la 7e cause de mortalité dans le monde.

On discerne plusieur types de diabètes :

— Le type 1 : Insulino-dépendant
— Le type 2 : forme plus bénigne ne nécéssitant pas d'injection d'insuline.
— Le diabète gestationnel.

Le projet que nous avons choisi de vous présenter porte sur le diabète de type 1 : les insulino-dépendant. Ce type de diabète est causé par le manque d'une molécule produite par le pancréas appellée insuline. Ce manque provient la plupart du temps d'une trouble auto-imunitaire qui détruit les cellules du pancréas produisant l'insuline. La présence d'insuline dans notre sang est essentielle. Cette molécule permet d'assimiler le sucre dans notre organisme afin d'apporter à nos organes et nos muscles l'énergie nécéssaire à leur bon fonctionnement.

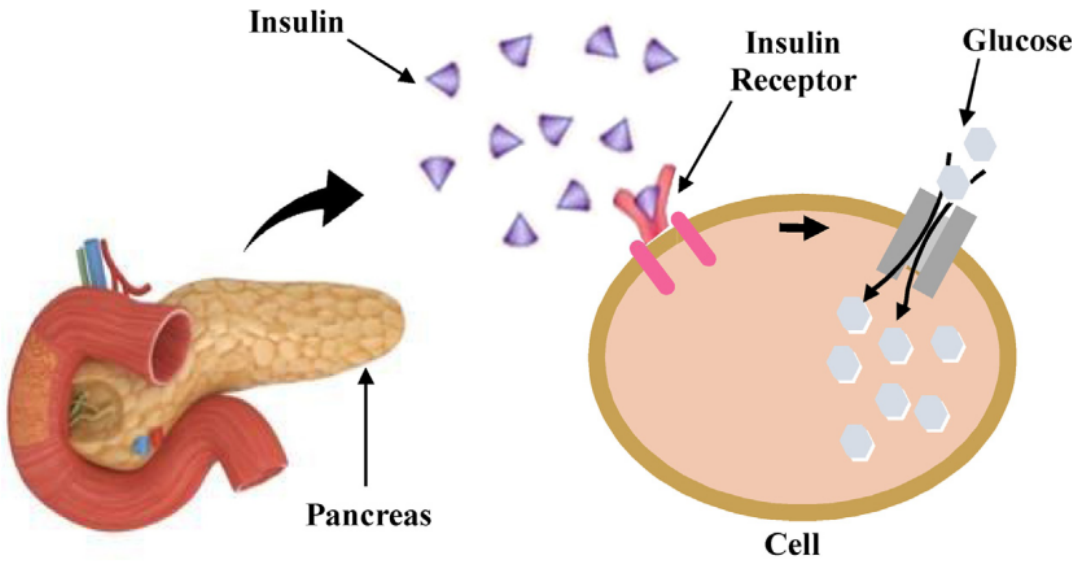

FIGURE 2 – Processus de l'absoption du glucose grâce à l'insuline

Le patient doit donc régulièrement contrôler son taux de glucose sanguin (TG) afin de ne pas avoir un taux anormal néfaste pour son organisme.

En effet, un taux de glucose sanguin supérieur à 150 mg/dL indique qu'une personne est en hyperglycémie. À l'inverse un TG inférieur à 70 mg/dL signifie que la personne est en hypoglycémie. Ces différents états doivent être à tout prix évités par le patient car ils causent généralement des séquelles à long terme. L'objectif est donc d'optimiser ses prises de glucides lorsqu'on est en manque de sucre (hypoglécémie) et d'insuline lorsqu'on a trop de sucre dans le sang (hyperglécémie). Cette gestion et ce suivit est possible grâce à un système de monitoring qui ne cesse d'évoluer d'année en année.

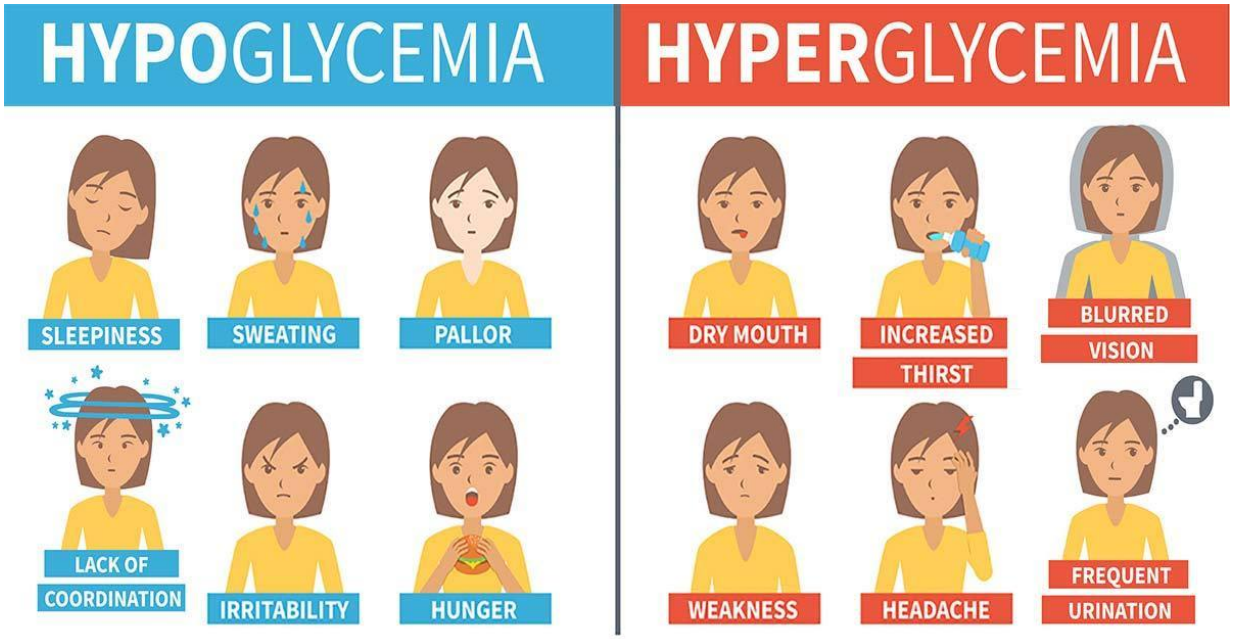

FIGURE 3 – Symptomes lors des différents états glycémiques

Quand une personne est diagnostiqué diabétique, un nouveau style de vie démarre pour lui. À partir du diagnostic le patient devra commencer à estimer la quantité de glucides dans les aliments qu'il ingère et s'injecter une dose d'insuline en conséquence qui lui permettra de bien métaboliser le sucre. Cet exercice demande beaucoup de rigueur et de reflexion. De plus, les systèmes de monitoring actuels ne sont pas capable de prédire à l'avance, sur un laps de temps plus ou moins long, un état critique pour un patient.

Notre projet vise donc à améliorer la qualité de vie des diabétiques de type 1 en facilitant leur gestion de leur taux glycémique grâce à la mise en place d'un algorithme prédictif annonçant un état dangereux pour le patient 15 minutes avant son arrivée. Ce laps de temps pourrait lui permettre de réagir correctement au problème car l'hyper/hypoglycémie entraine souvent des troubles de la concentration et un état de fatigue ne permettant pas d'y réagir avec discernement. De plus, une chute brutale et inattendue du taux peux entrainer un coma. Cela se produit surtout la nuit lorsque les patient ne sont pas attentifs à leur système de monitoring.

1.2 Le besoin d'un système de monitoring et de prédiction

Les systèmes de surveillance en continue du glucose sanguin (CGM) ont commencés à être développé dans les années 80 et arrivent sur le marché dès les années 2000 [?][?]. Il s'agit de système permettant de mesurer la concentration de glucose dans le tissu interstitiel comme on peut le voir sur la figure 5.

Le principe est de mesurer ce TG sur des intervals de temps régulier afin d'obtenir un graphique de ce taux en fonction du temps comme nous pouvons le constater sur la figure 4. Cette technologie a véritablement révolutionné la recherche sur le diabète en permettant de nouvelles possibilités sur l'analyses du diabète de type 1. En effet, il leur est possible maintenant de contrôler et réguler plus efficacement leur TG.

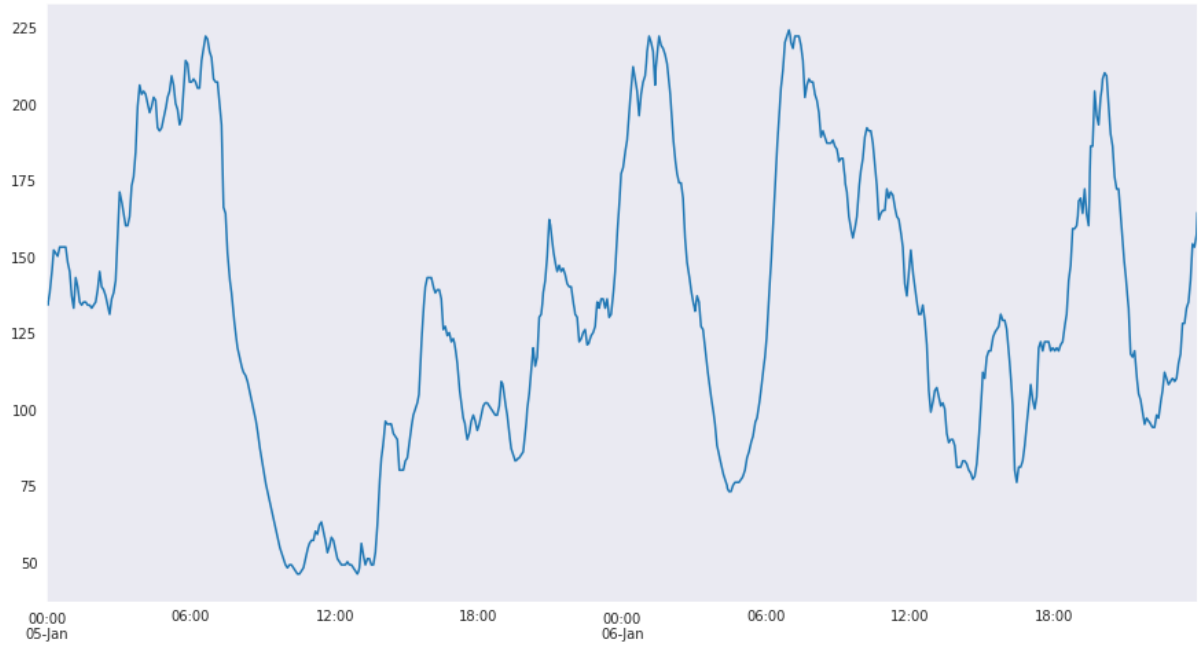

FIGURE 4 – Relevé du taux de glucose sanguin pour une période de 2 jours

Le CGM est composer d'un petit filament servant de capteur que le patient doit insérer sous sa peau. La mesure du GS se fait dans le liquide interstitiel pour être ensuite transmis au CGM qui analysera les données et en sortira une mesure

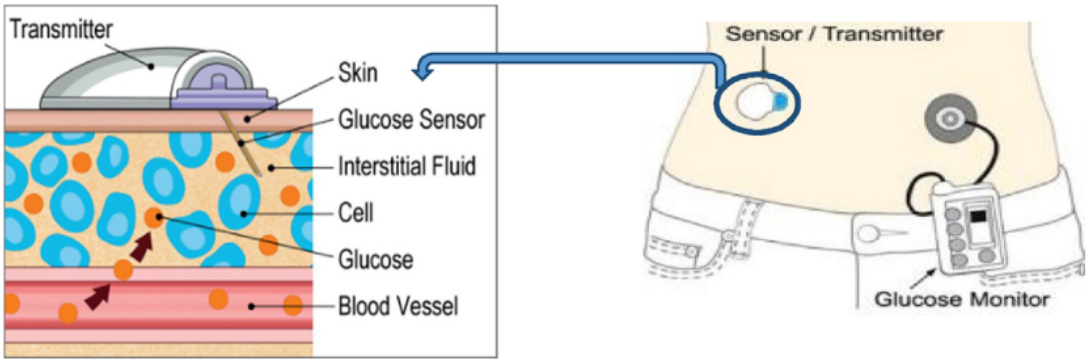

FIGURE 5 – Description du fonctionnement du système de monitoring

Concernant l'exactitude de la mesure de ces capteurs, il faut savoir qu'il y a un temps de latence entre le liquide interstitiel et le véritable glucose sanguin de l'ordre de 5 minutes.

Le capteurs va relever toutes les 10 secondes le TG actuel du sang et renvoyer une moyenne générale toutes les 5 minutes.

Ce système de monitoring nous permet donc d'obtenir des jeux de données conséquent sur des laps de temps court. En effet, comme nous venons de l'expliquer, le capteur mesure le GS toutes les 5 minutes ce qui signifie que sur une journée et pour un seul individus nous obtenons 288 relevés de données et approchons des 10.000 pour un mois ce qui semble relativement acceptable pour implémenter une méthode de machine learning.

1.3 Difficultés dans l'élaboration d'un algorithme prédictif du taux de glucose

Cela fait de nombreuses années que l'interêt ne cesse de grandir pour la prédiction du TG. De nombreux applications pourraient découler d'un succès dans la matière. Actuellement les systèmes de régulation de la maladie consiste a mesurer son TG grâce au capteur ou un glucomètre [1] et de s'injecter l'insuline directement dans le corps en conséquence. Cette pratique peut être nocive à long terme. En effet, le fait de se piquer régulièrement déclenche un mécanisme de défense de la part du corps qui va chercher à se protéger de cette perturbation extérieure.

Les nouveaux dispositifs plus élaborés tentent de lier les mesures du capteur avec une de pompe à insuline. L'objectif est de permettre à la pompe d'ajuster les quantités d'insuline qu'elle délivre grâce aux mesures instantanées du capteur. Ce dispositif s'approche d'une finalité dans la recherche pour la lutte contre le diabète : Le pancréas artificiel.

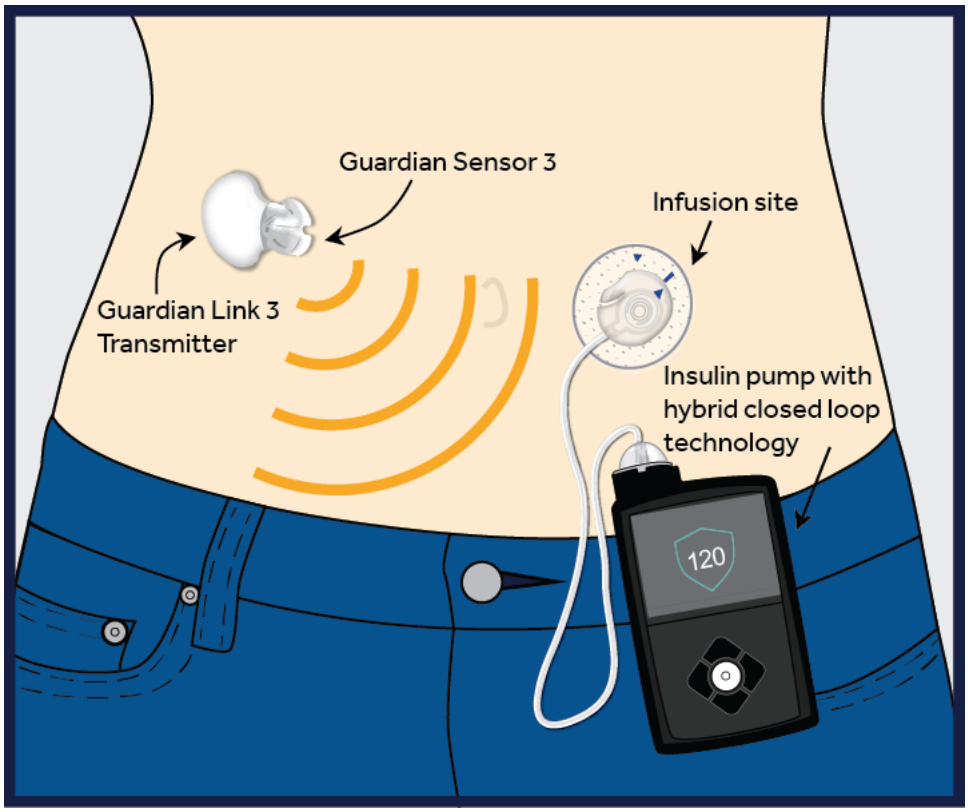

FIGURE 6 – Système de pancréas artificiel semi-autonome

1. appareil de mesure du TG nécessitant une goutte de sang du patient

Le pancréas artificiel consiste en un système capteur-algorithme-pompe qui optimise les quantités d'insuline injectées dans le corps grâce au relevés en temps réel du capteur.Ce procédé, plus performant que les simples systèmes capteurs-pompes, utilise un algorithme chargé de déterminer la tendance générale du TG ainsi que son intensité pour ordonner à la pompe d'injecter la dose suffisante d'insuline pour métaboliser correctement le glucose.

Même si le dispositif s'améliore et semble promis à un bel avenir, il reste un problème de taille entravant les espoirs des patients : le système ne peut prévenir sur un temps plus ou moins long les hypoglycémies. En effet, bien que la méthode soit très efficace pour prévenir des hyperglycémie, il n'en va pas de même pour son opposée l'hypoglycémie. Là où il suffit d'augmenter la quantité d'insuline pour éliminer le sucre en trop dans l'organisme, traiter une hypo demande l'action humaine. Lorsqu'un patient se retrouve en état d'hypoglycémie, la seule solution pour retrouver un taux normal est d'ingérer des aliments contenant suffisement de glucose (pain, soda, ...). C'est pourquoi ce système est considéré comme semi-autonome.

ce système est conçu pour agir instantanément en fonction des variations récentes du capteur, il n'est pas capable de prévenir le patient d'une hypoglycémie/hyperglycémie grave sur un interval de temps suffisent que pour que celui-ci puisse réagir correctement. En général, quand un diabétique se rend compte qu'il doit faire quelque chose pour sortir d'un état critique, celui-ci est déjà en train de subir les effets secondaire de celui-ci (fatigue, vision trouble, manque de coordination,...) . Ce qui peut mener à de problèmes plus graves.

Il est possible de retrouver une liste non-exhaustive de 42 facteurs pouvant impacter le TG. Dressé par Adam Brown dans son ouvrage [**?**]. Cette liste est reprise à la figure 7.

Commet vous pouvez le remarquer, cette liste est séparée en 6 catégories : la nourriture, la médication, l'activité ainsi que les facteurs environnementaux, biologiques et comportemental. Cette liste nous fait comprendre la difficulté d'avoir un modèle interprétant parfaitement tous les critères. Des études ont été réalisées et prennent en compte un maximum de ces critères afin de comparer leurs résultats avec des modèles purement mathématiques se basant uniquement sur les données précédentes des patients afin de déterminer si l'ajout du caractère physiologique permet d'améliorer les algorithmemes prédictifs [**?**] [**?**]. Nous discuterons de l'efficacité de ces modèles plus tard dans l'exposé.

Une autre difficulté émerge lorsqu'il est question de déterminer le TG chez un individus.En effet, chaque personne possède sa propre relation avec la maladie. Certain diabétique réagissent mieux que d'autre à l'insuline. Cette problématique ne nous permet pas de déterminer un modèle général capable de s'appliquer à tous les individus. La solution est donc de développer un algorithme capable de s'adapter en fonction de chaque individus. Un modèle de machine learning est donc parfaitement indiqué pour ce cas de figure car facilement personnalisable. De plus, cela permet de nous conforter dans le fait que l'étude d'un seul individu représente une bonne base pour entamer l'étude de notre projet. Si le modèle marche correctement sur un individu, nous essaierons de le mettre en défaut sur d'autres patients plus tard.

Enfin, pour réaliser ce projet, nous pouvons compter sur le soutiens du Dr. Isabelle Paris, endocrino-Diabétologue au CHR Mons-Hainaut.

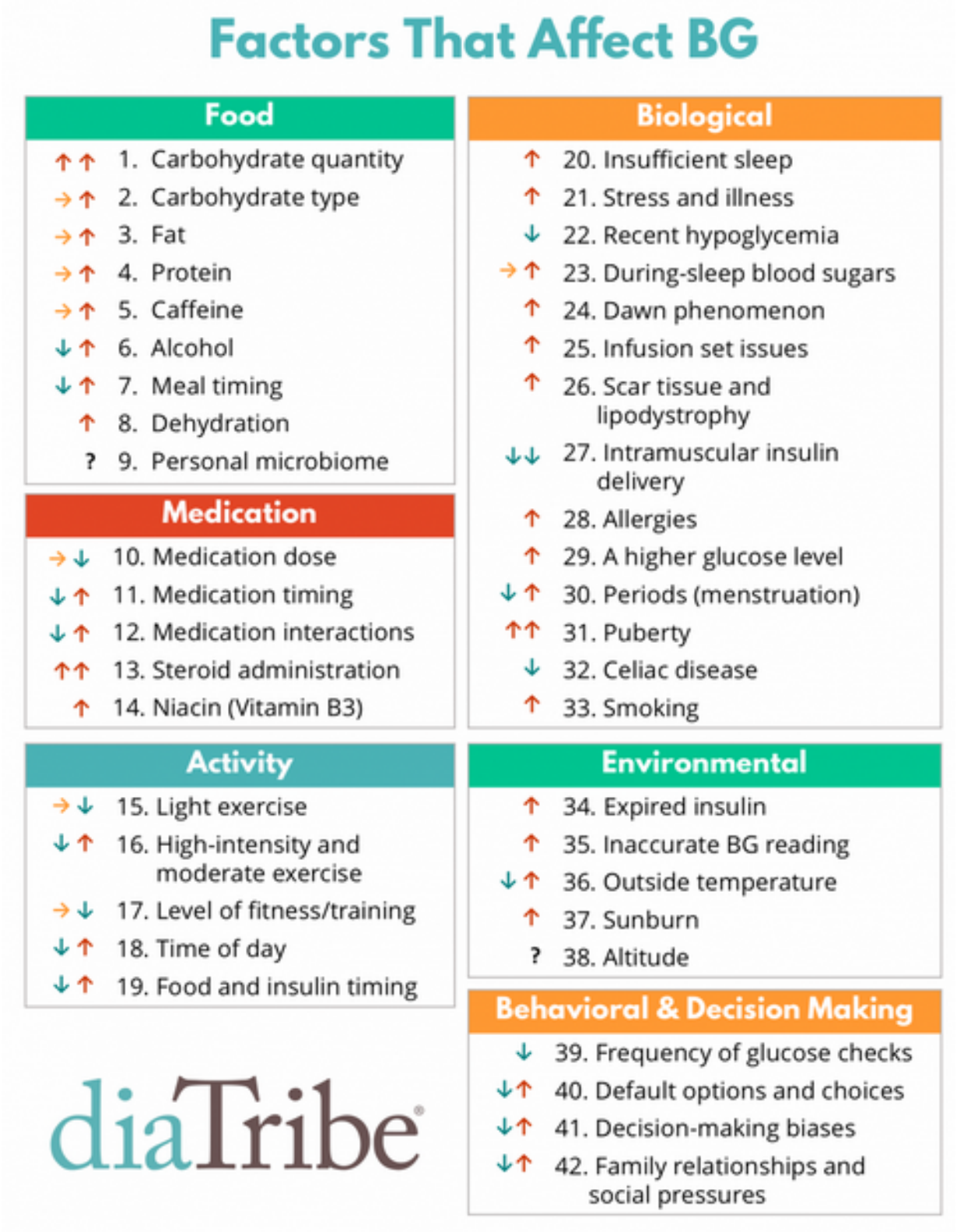

FIGURE 7 – Liste non-exhaustive des facteurs influençant le TG

2 Cahier des Charges

2.1 Recherche sur l'état de l'art

2.1.1 Méthode d'analyse temporelle

Les méthodes d'analyse temporelle sont des méthodes prédictives utilisant la temporalité et les évènememnts antérieurs afin de déterminer l'évolution d'un système donné.

Il existe de nombreux moyens pour essayer de prédire un état glycémique. Les recherches actuelles se concentrent sur des prédictions de l'ordre de 30' à 1h afin de permettre aux patients d'avoir un laps de temps convenable pour réagir. Ce que nous avons retenus des différents articles utilisés pour nous guider dans ce projet est que la plupart des algorithmes issus du machine learning sont plus performants que les prédictions des médecins spécialistes. Ce constat nous rassure sur notre démarche et nous pousse à nous renseigner d'autant plus sur le sujet.

2.1.2 Auto-regression

Un modèle d'auto-regression consiste en prédire des valeurs en fonctions d'un série de données temporelle (anciennes données). On appelle ordre d'une auto-regression, le nombre de valeurs précédentes de la séries qui ont été utilisées pour prédire la valeur à l'horizon de prediction (HP).

2.1.3 Modèle ARIMA

Il s'agit comme son nom l'indique d'un modèle de régression "classique" pour les séries temporelles. Le modèle peut être scindé en 2 parties : une part autorégressive (AR) et une part moyenne-mobile (MA).

Le modèle est généralement noté ARMA(p,q), avec p : l'ordre de la partie AR et q : l'ordre de la partie MA. Il s'agit d'un processus temporel discret vérifiant :

$$X_t = \epsilon_t + \sum_{i=0}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{1}$$

Il s'agit d'une méthode determinant un modèle sous-jacent pouvant prédire l'évolution du modèle en donnant un sens général au mouvement des données. Un gros défaut de cette méthode est qu'elle n'est utilisable que sur des données stationnaires et donc que le procédé ne possède aucune tendance haussière ou baissière. La moyenne du procédé doit donc être particulièrement stable ce qui n'est pas forcément le cas des jeux de données de l'étude. Ce qui explique donc la faible qualité de la méthode afin de prédire les BG futurs.

2.1.4 Prédiction par des médecins

L'article de l'université de l'Ohio sur l'approche du machine learning pour la prédiction du glucose sanguin [**?**] nous donne une bonne mesure sur la précision des prédictions données par des médecins spécialistes.

Horizon	t_0	ARIMA	Phys_1	Phys_2	Phys_3
30 min	27.5	22.9	**19.8**	21.2	34.1
60 min	43.8	42.2	**38.4**	40.0	47.0

FIGURE 8 – Erreur quadratique moyenne des prédictions des différentes méthodes

La colonne horizon fait référence à l'horizon de prédiction soit 30 ou 60 minutes. Le tableau nous permet donc de comparer les erreur quadratique moyenne (RMSE) de certaines méthode.
La première méthode, t0 sert de point de départ pour des comparaisons. Cette méthode triviale soutient le fait que le taux de glucose ne change jamais. Toutes ses prédiction amènent à la même valeur constante de TG. On peut donc forcément s'attendre à ce que cette méthode hasardeuse puisse avoir une RMSE élevée.

La méthode suivante est appelée Auto-Regressive Integrated Moving Average (ARIMA)(Box, Jenkins, and Reinsel 2008) et est expliquée au point précédent.

Ces résultats sont à comparer avec ceux de 3 docteurs, spécialistes du diabètes phys1,2 et 3
Le test de prédiction des médecins s'est effectué sur 5 patients T1D avec un ensemble de données d'évaluation de 200 points (40 par patient). Les médecins avaient accès aux données précédentes du patient afin de pouvoir composer leur propre généralisation de la fluctuation du glucose sanguin. Après cela, il leur a été demandé d'effectuer des prédictions pour 30 et 60 minutes. Après chaque prédiction la véritable valeur du BG leur était fournie afin qu'ils puissent affiner leur réflexion.

Plus loin dans l'article, un nouveau test est effectué mais cette fois ci grâce à une méthode de machine learning : le support vector regression (SVR). Comme nous le voyons sur la figure 9, le résultat est sans appel. Cette dernière méthode est bien plus performante que les 2 méthodes précédentes et que le meilleur résultat des médecins. Nous détaillerons le principe de la méthode SVR plus loin dans notre travail.

Horizon	t_0	ARIMA	Phys_1	SVRϕ	SVRϕ+A
30 min	27.5	22.9	19.8	19.6	**19.5**
60 min	43.8	42.2	38.4	36.1	**35.7**

FIGURE 9 – Erreur quadratique moyenne des prédictions des différentes méthodes

Il est à noter que le dernier cas présent sur la figure 9 est le cas d'un modèle SVR prennant en considération des facteurs physiologiques tel que le stress, la fatigue, la sensibilité à l'insuline, ... Même si ce dernier semble plus performant avec une RMSE légèrement plus faible, nous souhaitons opter pour un modèle prédictif non intrusif ou contraignant pour le patient. En effet, un tel modèle demande des entrées régulières de certaines données peu objectives ou quantifiables par le patient (Humeur, maladie,...) et ne constitue donc pas une voie exploitable pour l'amélioration du quotidien des personnes diabétiques.

Cette première approche des possibilités de réalisation d'algorithmes prédictif nous amène à considérer la méthode SVR très sérieusement pour nos propres recherches. En effet, cette méthode bassée sur la lecture de données antérieures afin de créer sa prédiction est très peu contraignante pour la patient mais reste tout de même à un bon niveau de performance.

2.1.5 Réseau de Neurone

Le *machine learning* a fait récemment un grand retour, puisque la loi de Moore (qui explique l'évolution de la puissance de calcul dans les ordinateurs commerciaux) finalement a rendu possible la mise en place de systèmes comme les réseaux de neurone. Parmi les avantages que ce type de modèle d'apprentissage automatique fournit se trouve : le traitement minimal nécessaire pour les données d'entrée et la possibilité d'analyser problèmes représentés dans un espace de grande dimension.
Pourtant, nous trouvons que parfois ce type de modèle n'arrive pas à apprendre comme il devrait le faire idéalement. C'est-à-dire, même si la valeur du R^2 est haut sur l'ensemble d'entraînement, la qualité des prédictions sur les données inconnues est mauvaise, indiquant que l'algorithme n'a pas su généraliser un modèle à partir des tendances observées dans les exemples fournis. [**?**]

```
R^2 and Loss at last iteration of each nn (on the TRAINING set):
(10, 5, 8)        = 0.6135271136952117
                  = 436.2725005061999

(8, 6, 8)         = 0.7519083160390774
                  = 281.1777795297427

(10, 6, 4)        = 0.8226195992135606
                  = 202.12640125472507

(8, 5, 3, 2)      = -1.5515238532382014e-05
                  = 1120.6967557275123

(9, 6, 4, 2)      = 0.7146650567265362
                  = 323.65697516939707

(20, 20)          = 0.8946222778179591
                  = 120.30421727934714

(15, 5, 5)        = 0.7785498897395353
                  = 247.87846173956822

(5, 5, 3)         = 0.6030796863499214
                  = 446.46820840005626
```

(a) Ensemble d'entraînement

```
R^2 and Loss at last iteration of each nn (on the TESTING set):
(10, 5, 8)        = -0.036514618445658176
                  = 436.2725005061999

(8, 6, 8)         = -0.29513673829517817
                  = 281.1777795297427

(10, 6, 4)        = -0.17882975895838937
                  = 202.12640125472507

(8, 5, 3, 2)      = -0.4074895119043671
                  = 1120.6967557275123

(9, 6, 4, 2)      = -0.0825100716892333
                  = 323.65697516939707

(20, 20)          = -0.3203596997295073
                  = 120.30421727934714

(15, 5, 5)        = -0.3314294074492463
                  = 247.87846173956822

(5, 5, 3)         = 0.006622496474008144
                  = 446.46820840005626
```

(b) Ensemble de vérification

FIGURE 10 – Précision des Réseaux de neurone

Comme on peut voir dans la figure 10, les modèles travaillés par G. Magaña sous la supervision du Dr. M. Heinen [?], arrivent à imiter le comportement de l'évolution glycémique de l'ensemble de méditions pendant les jours observés. Pourtant, ils ne peuvent pas prédire le comportement des jours suivants non observés dans l'intervalle d'entraînement.
Le modèle était simple : En fonction de la glycémie des heures précédentes, quantité de glucides ingérée et dose d'insuline administrée ; le réseau de neurone devait pouvoir prédire la glycémie des heures suivantes.

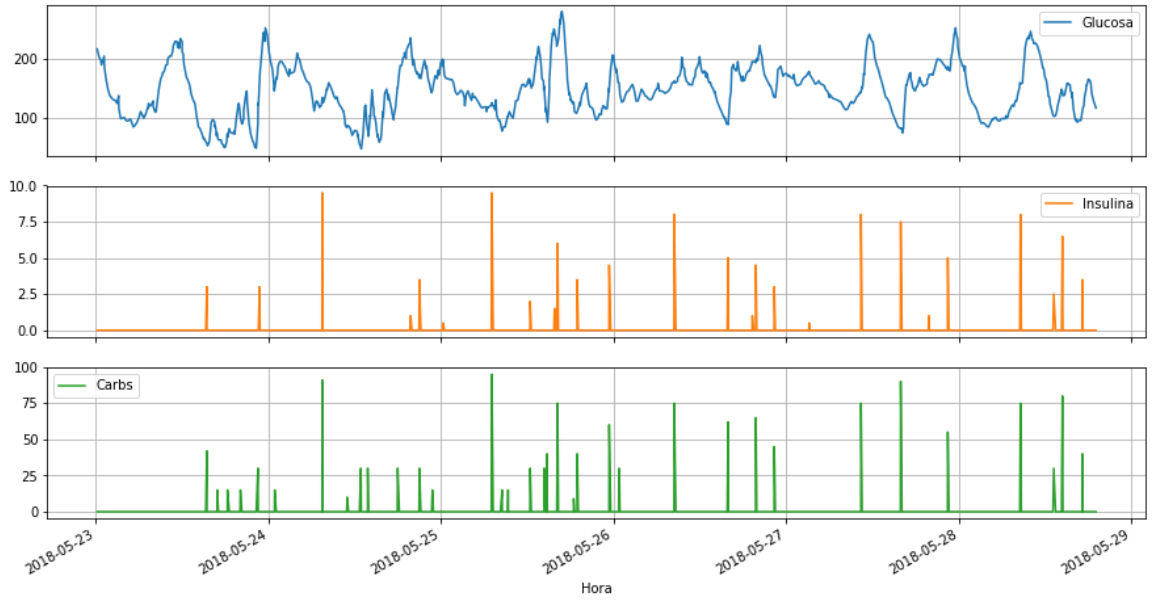

FIGURE 11 – Vue de la forme des données fournies au modèle de prédiction de l'évolution glycémique.

Ils attendaient que le réseau de neurone soit capable d'imiter le comportement du métabolisme glycémique humain. Ça voudrait dire que le modèle prédirait une glycémie plus basse si on augmentait la dose d'insuline entrant dans le même système. Ce comportement peut être observé dans la figure 11.

RN évalué sur l'ensemble d'entraînement

Pour toutes les figures suivantes, la courbe verte représente l'ensemble des données réelles, la courbe orange la moyenne mobile et les points bleus les prédictions du Réseau de neurone. On peut voir que plus de neurones dans chaque couche améliorent la précision de la prédiction mais plus de couches ne le font pas nécessairement.

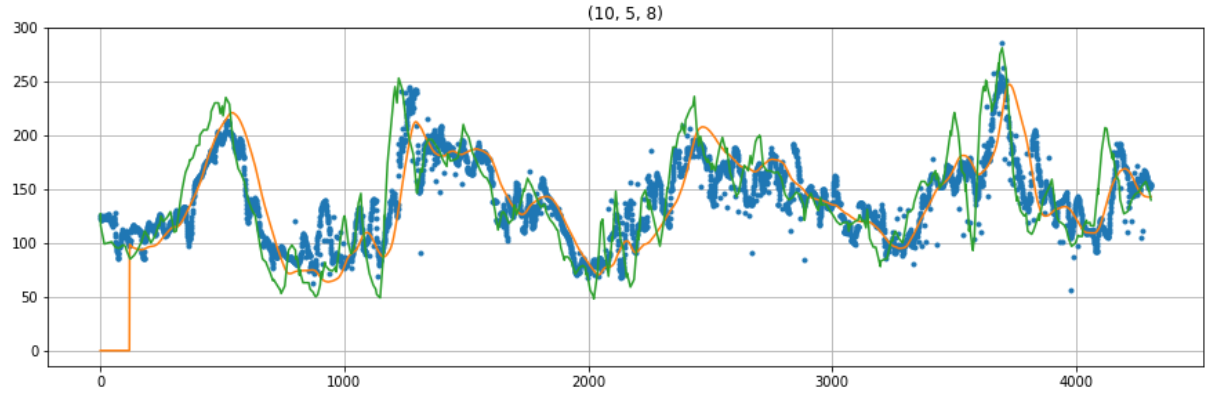

FIGURE 12 – Entraînement trois couches : 10, 5, 8

La première configuration (figure 12) a un certain décalage, visible surtout près des montées de pente considérable. Dans ce cas, la qualité de la prédiction n'est pas meilleure que celle de la moyenne mobile, un des outils plus simples pour analyser une série temporelle. Nous pouvons aussi constater une dualité des comportements opposés : D'un côté la moyenne mobile fait disparaître des oscillations et le bruit en accentuent la tendance générale, tandis que le RN essaie d'imiter le comportement instant par instant. Ce dernier objectif fait apparaître quelques irrégularités.

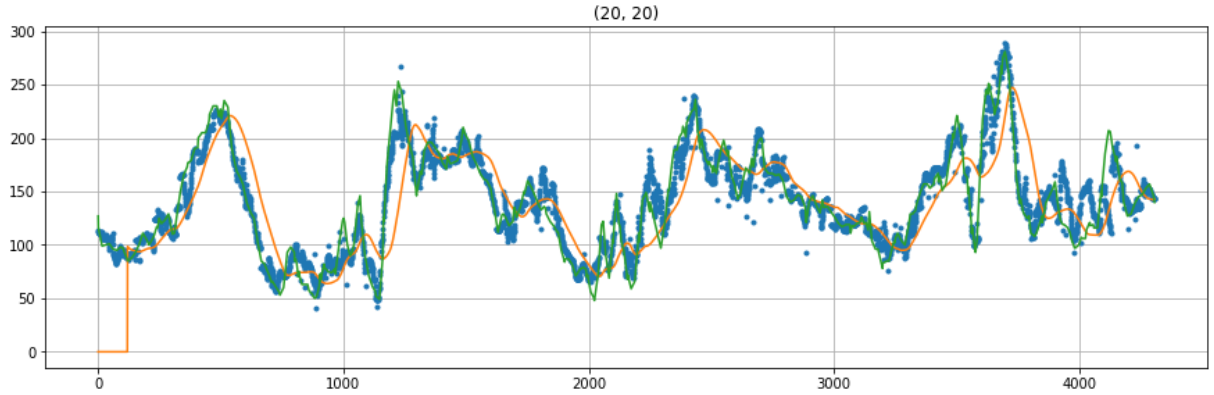

FIGURE 13 – Entraînement deux couches : 20, 20

Une seconde configuration, (figure 13) ne compte qu'avec deux couches internes, chacune composée de 20 neurones. Ce modèle arrive un niveau de prédiction d'haute précision : Les points que l'on peut voir sur la figure collent très bien à la ligne définie par les données. Ce modèle a vraiment minimisé l'erreur, c'est-à-dire la différence entre la valeur observée et celle produite en fonction des données précédentes trouvées dans l'historique.

D'après les courbes observées l'erreur moyen du modèle réseau de neurones est inférieur à celui de la moyenne mobile. De cette précision on s'attendrait à que ce modèle soit optimale pour prédire l'évolution glycémique. Pourtant, nous verrons dans la suite de ce travail que ce que l'on observe ici concerne un phenomène connu sous le nom de surapprentissage ("overfitting" en anglais)

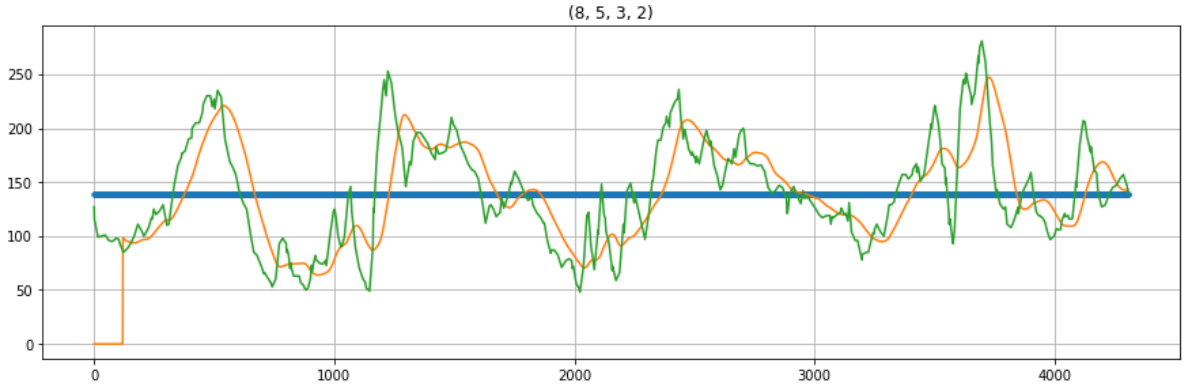

FIGURE 14 – Entraînement quatre couches : 8, 5, 3, 2

Dans la figure 14 on peut remarquer qu' augmenter le nombre de couches cachées empêche le modèle de minimiser (diminuir en tout cas) l'erreur. Ceci finit par le rendre inutilisable. Rappelons qu'ici il s'agit d'un rejet déjà au niveau de l'entraînement. Réussir cette étape ne veut pas dire que le modèle est acceptable ou optimale. Pour qu'il soit acceptable, une bonne précision sur l'ensemble d'entraînement est une condition nécessaire mais pas suffisante.

RN évalué sur l'ensemble de vérification

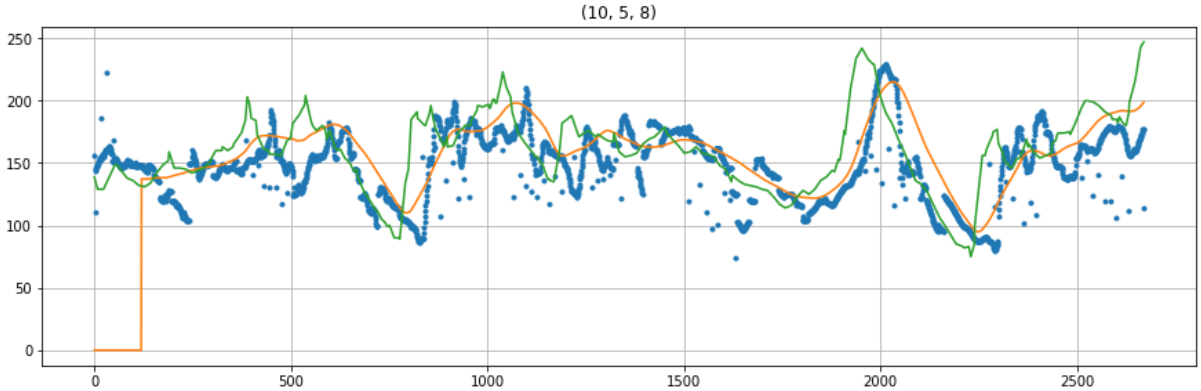

FIGURE 15 – Vérification trois couches : 10, 5, 8

La première configuration (figure 15) qui montrait déjà un certain décalage, visible surtout près des montées remet des résultats encore moins précis dans l'ensemble de vérification que dans l'ensemble d'entraînement. Même si ce comportement est attendu dans la plupart des cas, la qualité de la prédiction est minimale vu que le modèle prédit des montées et descentes lorsqu'il n'y en a pas. Donné ce nombre de faux positives pour les piques, la valeur de cette configuration est remise en question. Pourtant la moyenne mobile, avec son décalage certain et le fait qu'elle rend plus lisse le comportement local et globalement, maintient la précision fournie sur les deux ensembles. Ce n'est pas étonnant de voir comment un modèle qui avait des défauts clairs sur l'ensemble d'entraînement diminue sa performance dans l'étape de vérification, étant exposé à des données complètement inconnues. Pourtant, en regardant la performance du réseau avec deux couches cachées de 20 neurones chacune, on s'attendrait à une performance meilleure de celle que l'on obtient.

Pourquoi alors, obtient-on le comportement erratique observé dans la figure 16 ? C'est dû au surapprentissage. Si l'on compare à une personne qui étudie n'importe quel sujet, ça searit l'équivalent de mémoriser tous les exemples vus sans arriver à une généralisation du comportement ou concept. Dans la figure 17 on peut remarquer qu'augmenter le nombre de couches cachées (ce qui avait empêché le modèle de minimiser l'erreur) n'est pas utile dans l'ensemble de vérification.

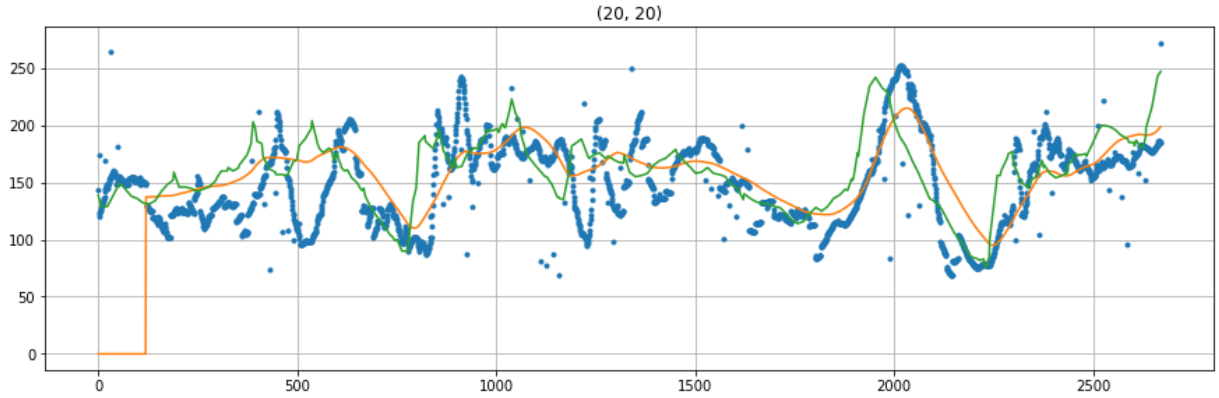

FIGURE 16 – vérification deux couches : 20, 20

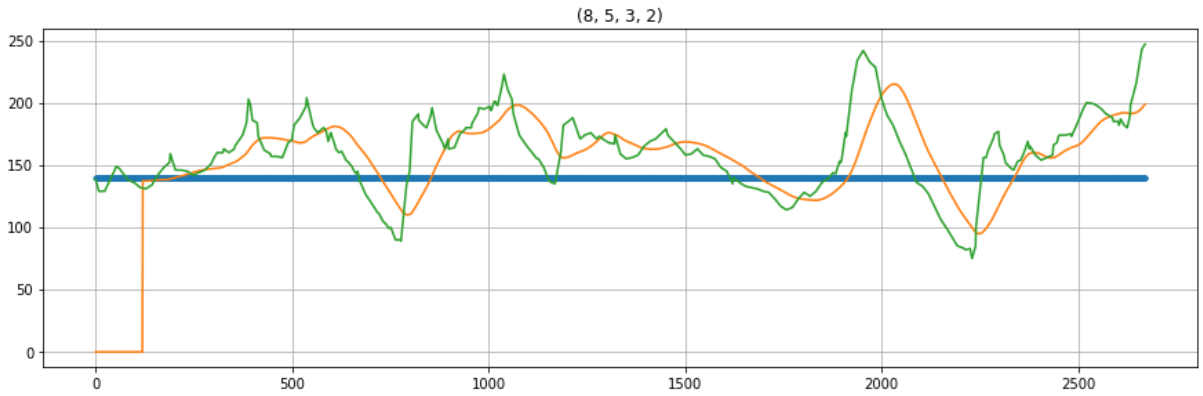

FIGURE 17 – Vérification quatre couches : 8, 5, 3, 2

Une dernière épreuve pour ce modèle de réseau de neurones a été observé sur la façon dont il réagissait à des modifications dans les donnés d'entrée. Est-ce qu'augmenter le taux basal d'insuline [2] prédit une réduction globale de la glycémie ? Si le modèle imitait fidèlement le métabolisme glycémique ça serait le cas, pourtant il ne l'est pas. Quand Magaña et Heinen ont regardé la réponse du système à une augmentation globale du taux basal d'insuline, les valeurs prédites furent contraire à ce qu'ils attendaient : elles avaient monté à la place de baisser. Ceci nous permet de conclure que le réseaux de neurones ne sont pas optimaux pour prédire l'évolution glycémique.

2.1.6 Support vector regression

La méthode de régression par vecteur de support est l'une des technique les plus utilisées dans le domaine de l'analyse prédictive de données car elle amène à de très bons résultats [?] [?] [?]. Ce modèle, très proche de celui des machines à vecteur de support (SVM) tient sa principale différence dans le fait que le SVR renvoie une donnée numérique là où le SVM permet seulement de prédire un état/catégorie.

2. quantité d'insuline nécessaire pour apporter la quantité de base de glucose à notre organisme

L'objectif du SVR est de déterminer un modèle (régression) capable de prédire avec la plus grande précision l'évolution d'une série temporelle pour une période appelée horizon de prédiction (PH).

Comme la méthode SVR peut potentiellement s'appliquer sur un jeu de donnée non linéaire, comme dans notre cas, la fonction de prédiction doit prendre la forme suivante :

$$f(x) = w^T \phi(x) + b \tag{2}$$

Où $\phi(x)$ est une fonction transformant les données en un espace de fonctions de dimension supérieure pour permettre la séparation linéaire, w est la matrice de poids et b le biais.

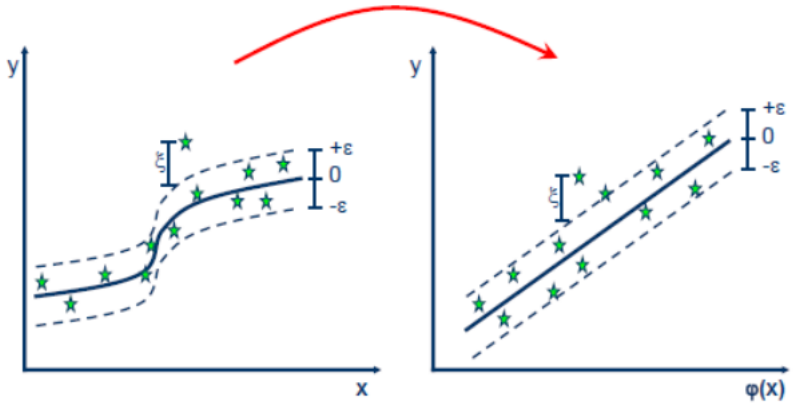

FIGURE 18 – Linéarisation de l'ensemble de données de base

Comme expliqué précédemment, l'algorithme cherche à résoudre un problème de régression non linéaire en projettant les données d'apprentissage dans un nouvel espace ϕ où la relation entre l'entrée x_i et la sortie y_i devient linéaire.

Trouver la fonction qui modélisera la série temporelle revient à résoudre le problème d'optimisation suivant :

$$min\ \frac{1}{2}\ ||w||^2 + C\ \sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{3}$$

La constante C est un coéficient représentant le lien entre l'uniformité de la fonction SVR et les écarts supérieurs à ϵ tolérés.

$$\begin{cases} y_i \leq f(x_i) + \epsilon + \xi_i \\ y_i \geq f(x_i) - \epsilon - \xi_i \\ \xi_i \xi_i^i \geq 0 \end{cases} \tag{4}$$

D'un point de vue mathématique, le SVR tend à réduire l'erreur tolérée en maximisant la marge de l'hyper-plan séparant les données. Cet hyperplan peut-être déterminé grâce à un SVM. Une zone est d'abord définie entre $f(x) + \epsilon$ et $f(x) - \epsilon$. Cette zone appelée $\epsilon - insensitive$ et définie la limite à partir de laquelle l'erreur commencera à augmenter linéairement. Cela signifie que la distance à l'hyperplan pour les données présentes dans la zone ne seront pas comptabilisés comme erreur. Par contre, tout point en dehors de la zone aura pour impacte d'augmenter l'erreur proportionnellement à la distance du point avec l'hyperplan. Les données en dehors de la zone $\epsilon - insensitive$ servent donc de vecteur de support pour le modèle.

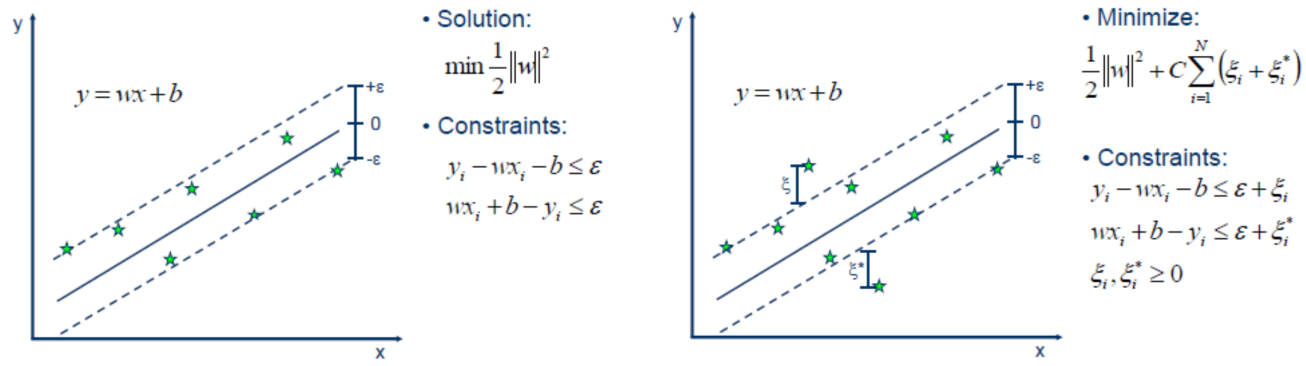

FIGURE 19 – Principe de la régression par vecteur de support

Enfin, la dernière chose à prendre en compte est le choix du noyaux de fonction pour notre algorithme. Pour notre travail, nous utilisons la fonction à base radiale ou RBF (radial basis function) définie par :

$k(x_i, x_j) = exp\left(\frac{-||x_i - x_j||^2}{2\gamma^2}\right)$

Où γ est le paramètre du noyaux qui doit toujours être plus grand que 0.

Comme le système d'équations (4) remplis les conditions KKT (Karush-Kuhn-Tucker) [**?**], la fonciton de régression peut être utilisée pour prédire un nouveau x de sorte que :

$$f(x) = \sum_{i=1}^{N} (a_i - a_i^*)k(x, x^i) + b \tag{5}$$

Avec $(a_i - a_i^*)$ comme multiplicateur de Lagrange et $k(x, x^i)$ la fonction de noyaux

2.1.7 Discussion sur l'ajout d'un modèle physiologique

La discussion sur l'ajout d'un modèle physiologique aux modèles mathématiques semble prépondérante. En effet, bien que les algorithmes basés sur l'étude des donnés précédentes des patients semble assez précis, ceux-ci ne prennent pas en considerations les nombreux facteurs physiologique faisant varier le taux de glucose. Cette négligence semble dans un premier temps être la cause d'une limite dans les performances des algorithmes prédictifs.

2.1.8 Mesures de performances des modèles de régression

MAPE et RMSE

2.2 Outils

Afin de faciliter le travail collaboratif et le suivis du projet, nous avons décidé de créer un répertoire sur Github :

`https://github.com/gmagannaDevelop/DiabManager.git`

Ce repertoire nous permettra de continuer le travail entamé à l'avenir et le partager avec d'autres personnes souhaitant également y contribuer. Pour compiler notre code Python, nous utilisons Jupyter programme très efficace pour notre projet. Cela nous permet de voir les graphiques et résultats pour chaque étape du code.

2.3 Objectif 1 : Modèles de classification

Principal : Modèle de classification

Même avec la formation donnée par les médecins, diététiciens, etc il existe toujours une certaine incertitude chez le patient : Est-ce que cette combinaison de facteurs va amener mon taux de sucre vers le bas ou vers le haut ? Dit autrement : Ce que je veux manger ou cette activité physique que je suis en train de pratiquer ferra-t-elle en sorte que je tombe en hypo ou hyperglycémie ? Les modèles de prédiction intégrés dans les pompes à insuline ne prennent en compte que le taux actuel du sucre et la tendance.

Afin d'améliorer la valeur de notre algorithme, surtout au niveau thérapeutique, la prédiction se fera dans un premier temps sous la forme d'un modèle de classification plutôt que de régression. Tout cela afin d'avoir une première approche du machine learning. Une fois que nous aurons bien compris le fonctionnement, nous implémenterons un modèle de regression.

2.3.1 Support vector machine

Les machines à vecteur de support sont des méthodes de classification. Le principe est de créer un hyperplan séparant les données en plusieurs groupes. Son procédé est décrit plus en détail au point 3.1.

2.4 Objectif 2 : Modèle de régression

2.4.1 Support vector regression

Notre second objectif consiste à réaliser un modèle de régression afin d'obtenir une donnée chiffrée beaucoup plus utile pour le patient. En effet, connaitre la gravité d'un état grâce à une valeur chiffrée lui permettra de mettre en place un dispositif de contre suffisent. La méthode SVR a déjà été décrite précédemment. Nous l'utiliserons dans la suite de notre travail car elle représente l'une des meilleurs méthode actuelle pour la prédiction du TG.

3 Solutions et résultats

3.1 Résultats du SVM

Ici quand on parle de SVM, on parle de la mise en place des machines à vecteurs de support en tant que classeurs. Dans la litterature on peut aussi le trouver le nom SVC (qui veut dire Support Vector Classifier en anglais). Le modèle que l'on a proposé était simple et efficace : Il nous a permis de commencer à explorer la précision des modèles sur un jeu de données de taille réduite et sans besoin de beaucoup de temps ou puissance de calcul pour l'entraînement.

Les SVMs peuvent travailler bien et fournir des prédictions de qualité même avec un jeu de données de taille réduite. Comme au début on n'avait pas accès à la totalité des données du capteur stockées dans la pompe à insuline, nous avons développé une application mobile en Python pour obtenir quelques données.

```
In [31]: data.head()
```

Out[31]:

dateTime	activeInsulin	carbs	insulin	trend	glycaemia	hour	tag	postp tag
2019-02-07 09:24:00	0.0	30.0	3.75	0.0	184.0	9	hyper	hyper
2019-02-07 12:53:00	0.0	56.0	4.95	0.0	133.0	12	normo	hyper
2019-02-07 17:39:00	0.0	12.0	1.00	0.0	99.0	17	normo	normo
2019-02-07 18:13:00	0.0	10.0	0.00	0.0	101.0	18	normo	normo
2019-02-09 10:52:00	0.0	85.0	7.80	0.0	105.0	10	normo	hyper

FIGURE 20 – Forme des données utilisées pour entraîner le modèle de prédiction comme classification

Dans les modèles de machine learning, souvent on trouve des hyperparamètres qui doivent être fixés avant d'entraîner le modèle et qui malheureusement on un impact très fort sur la performance globale. Un outil qui nous permettre de chercher une valeur optimale pour tous les hyperparamètres d'un modèle est le Grid Search (recherche en grille, en français). La librairie SciKit-Learn possède une fonction pour faire le GridSearch qui permet d'explorer la précision d'un modèle en itérant sur un ensemble d'ensembles de valeurs possibles pour chaque paramètre.

Grid Search for optimizing parameters

```
In [44]: param_grid = {
             'kernel': ['linear', 'rbf', 'sigmoid', 'poly'],
             'C': [0.1, 1, 10, 100],
             'gamma': [1, 0.1, 0.01, 0.001, 0.00001, 10]
         }

In [53]: # Make grid search classifier
         clf_grid = GridSearchCV(SVC(), param_grid, verbose=1, cv=10)

In [54]: clf_grid.fit(X_train, y_train)

         Fitting 10 folds for each of 96 candidates, totalling 960 fits
```

FIGURE 21 – Grille des valeurs pour les hyperparamètres

Une fois que l'on a obtenu une précision moyenne de 0,767 on s'est rendu compte qu'ils s'agissait bien d'un type de problème linéairement séparable et que les machines à vecteurs de support étaient une bonne méthode pour analyser le problème.

La classification

3.1.1 Grid Search

Pour construire un bon modèle de SVR il faut trouver les paramètres γ, C et ϵ qui minimiseraient les erreurs.

C'est justement l'objectif de la grid search. Ce processus cherche à trouver les hyperparamètres dans le but de trouver les valeurs optimales d'un modèle donné. Le fonctionnement est simple. L'algorithme va tester toutes les valeurs de paramètre possible dans des intervalles donnés et ainsi retourner la meilleure combinaison.

```
In [63]: print(f'Cross validation scores: \nMin: {optimum.min()}, Mean: {round(optimum.mean(),3)}, Max: {optimum.max()}')

Cross validation scores:
Min: 0.25, Mean: 0.767, Max: 1.0

In [64]: sb.distplot(sigmoid_cvl)

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x7f20f2d12b70>
```

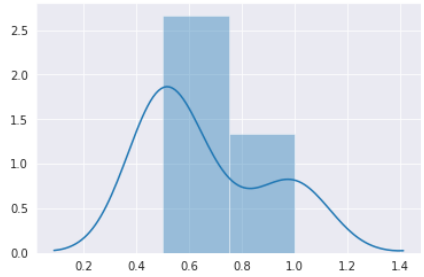

FIGURE 22 – Description des résultats de la validation croisée

```
In [247]: GS.GridSearch(param_grid=params)

Fitting 10 folds for each of 72 candidates, totalling 720 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:   14.5s
[Parallel(n_jobs=-1)]: Done 192 tasks      | elapsed:  1.1min
[Parallel(n_jobs=-1)]: Done 442 tasks      | elapsed:  2.6min
[Parallel(n_jobs=-1)]: Done 720 out of 720 | elapsed:  6.2min finished
/home/gml/anaconda3/envs/DiabManager/lib/python3.6/site-packages/sklearn/model_selection/_search.py:841: DeprecationWarn
ing: The default of the `iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This
will change numeric results when test-set sizes are unequal.
  DeprecationWarning)

Best Parameters:
 {'C': 100, 'epsilon': 0.01, 'gamma': 0.001, 'kernel': 'rbf'}
Best Estimators:
 SVR(C=100, cache_size=200, coef0=0.0, degree=3, epsilon=0.01, gamma=0.001,
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

FIGURE 23 – Paramètres déterminés par notre grid search

Ce processus est indispensable car les performances du modèle dépendent des hyperparamètres déterminés par celui-ci. Dans la figure 23, On peut observer que les valeurs optimales des hyperparamètres du model sont :

— C = 100
— $\epsilon = 0.01$
— $\gamma = 0.001$
— kernel = "rbf"

La suite de notre implémentation du SVR s'appuiera sur les valeurs de ces paramètres

3.2 Résultat du SVR

3.2.1 Comparaison avec de précédents travaux

Afin de pouvoir vérifier l'éfficacité de notre GS, nous avons décidé de comparer nos résultats avec et sans ce système d'optimisation. Le modèle a été entrainé sur un jeu de donnée de ... jours et testé sur une durée de ... jours. Les valeurs des hypperparamètres sont celles proposées par le site sci-kit learn à titre d'exemple.

On observe que sans la GS, la RMSE (root mean square error) est très élevée ce qui indique que le modèle n'est guère fiable et manque de précision. Pour avoir la RMSE il suffit de prendre la racine carré de l'erreur quadratique moyenne obtenue sur la figure 24. On observe que pour un jeu de données inconnues (échantillon de test) la RMSE est de : $\sqrt{272} = 16{,}49$ ce qui correspond à un niveau très moyen pour un algorithme de prédiction. Nous avons également vérifié les performances du modèle en le confrontant au set de donnée utilisé pour l'entrainement du modèle. Comme on peut s'y attendre la RMSE pour ce jeu de donnée chute drastiquement pour obtenir : $\sqrt{70,77} = 8{,}41$. Ce qui correspond à un excellent score.

```
In [190]: Z = SVRegressor()

In [191]: Z.load_model(filename='models/rbf_21days_tseries_model.sav', kernel='rbf')

In [200]: mean_squared_error(split(y2, 2)[0], Z.predict(kernel='rbf', X=split(X2, 2)[0]))
Out[200]: 271.51587514286445

In [204]: mean_squared_error(y, Z.predict(kernel='rbf', X=X))
Out[204]: 70.77212423591848

In [201]: mean_squared_error(y2, Z.predict(kernel='rbf', X=X2))
Out[201]: 272.0313331000548
```

FIGURE 24 – Résultats pour un modèle sans grid search

Voici maintenant nos résultats pour le modèle implémenté à partir des hyperparamètres calculés par notre grid search :

```
In [343]: GS.MAPE(y.reshape(-1, 1), GS.predict(kernel='rbf').reshape(-1, 1))
Out[343]: 5.091708442922069

In [345]: GS.MSE(X=X, y=y, kernel='rbf')
Out[345]: 103.6163856291215

In [348]: GS.MAPE(y2.reshape(-1, 1), GS.predict(kernel='rbf', X=X2).reshape(-1, 1))
Out[348]: 7.240546763261388

In [349]: GS.MSE(X=X2, y=y2, kernel='rbf')
Out[349]: 195.15848694194662
```

FIGURE 25 – Résultats pour un modèle avec grid search

Comme on peut le voir sur la figure 25, les résultats pour la méthode avec grid search sont bien meilleurs sauf peut-être sur les données d'entrainement. Cela est du au fait que l'algorithme cherche à être le plus gobale possible et n'est pas spécialement préparé qu'avec les données d'entrainement au vues des hyperparamètres qui ont été calculés par le grid search. Pour les données d'entrainement nous avons une RMSE = 10,18 qui est un très bon résultat. Pour la partie qui nous interesse le plus, le modèle inconnu par l'algorithme, celui ci obtient une RMSE = 13,97 ce qui correspond à une amélioration de la RMSE de 2,52. Nous avons donc bien réussi l'optimisation de notre modèle.

Ces données ont été prises pour des prédictions de l'ordre de 15'. Pour comparer nos résultats, nous devons tester le modèle sur le même horizon de prédiction que les autres études.

```
In [49]: GS.MAPE(y.reshape(-1, 1), GS.predict(kernel='rbf').reshape(-1, 1))
Out[49]: 10.63068703903339

In [50]: GS.MSE(X=X, y=y, kernel='rbf')
Out[50]: 413.1018779493761

In [51]: GS.MAPE(y2.reshape(-1, 1), GS.predict(kernel='rbf', X=X2).reshape(-1, 1))
Out[51]: 10.547511747032642

In [52]: GS.MSE(X=X2, y=y2, kernel='rbf')
Out[52]: 231.91784803400336
```

FIGURE 26 – Résultats pour un modèle avec grid search sur un HP de 30'

Notre modèle a été entraîné sur 10 jours de données précédentes et testé sur 4 jours comme dans l'étude avec laquelle nous comparons nos résultats. Comme on peut le voir, nous obtenons une RMSE de 20,32 ce qui correspond à une très bonne performance meilleure que t0, ARIMA et 2 des 3 diabétologues. De plus, nous sommes assez proche du résultat pour le SVR de l'étude

(19,6). Cette différence est sans aucun doute due au fait que nous n'ayons qu'un seul individu sur qui expérimenter.

4 Perspectives d'améliorations

4.1 Ajout de certains paramètres physiologique non contraingnant au SVR

Nous avons vu précédemment que l'ajout d'un modèle physiologique était très compliqué à mettre en place car le patient devait régulièrement entrer ses paramètres phisiologiques parfois très subjectifs (humeur, niveau de stress,...) pour un faible rendement en terme de performance. Pourtant, une piste interessante pour l'avenir serait de chercher parmis ces facteurs ceux ne représentant pas une contrainte et une régularité dans son suivis. Par exemple, il pourrait être interessant de prendre en considération la résistance à l'insuline ou la sensibilité à l'insuline du patient. Ces facteurs varient très peu et peuvent être facilement calculés par les diabétologues.Cela permettrai d'améliorer les modèles mathématique sans perdre leur avantage non intrusif pour le patient.

4.2 Implémentation à une platforme mobile

Le fait que l'algorithme fonctionne différemment d'un réseau de neuronne lui donne un gros avantage sur sa transportabilité. En effet, le SVR a besoin de beaucoup moins de puissance de calcul qu'un réseau de neuronne ce qui limite l'encombrement utile. Certaines études se sont déjà portées sur le sujet [**?**] dans le but d'utiliser le SVR sur des plateformes mobiles.

Conclusion

Finalement, après comparaison avec les autres études, nous pouvons faire une première conclusion sur la forte probabilité que l'algorithme que nous avons implémenté est fiable. Il possède de très bonnes performances en dépassant les prédiction des médecins mais est à critiquer pour son manque de tests. En effet, nous n'avons testé les données que sur une seule personne ce qui ne nous permet pas d'être certain à 100 pourcent de son efficacité. A l'avenir, nous testerons notre algorithme sur d'autres patients pour confirmer la validité de notre recherche.

Annexes

Références