

# Prédiction de temps de parcours de bus par chaînage des données d'entraînement

Gabriel Ferrettini, François Queyroi, Mounira Harzallah

Nantes Université, Ecole Centrale Nantes, CNRS, LS2N, UMR 6004  
F-44000 Nantes, France  
prenom.nom@univ-nantes.fr

**Résumé.** Dans le cadre de la planification de lignes de bus, il est intéressant de prédire à la fois la durée d'une course (du départ au terminus) et le temps de parcours de chaque inter-arrêt. Une solution est de chaîner les prédictions de modèles construits pour chaque inter-arrêt et de les sommer pour obtenir le temps total de la course. Cependant, ce modèle dit "chaîné" ne prend pas en compte la dégradation des données prédites d'une étape pour l'étape suivante. Nous proposons d'améliorer ce modèle en intégrant le chaînage des prédictions dans l'entraînement des modèles de chaque étape. L'évaluation de notre proposition montre que notre méthode améliore la prédiction de la durée totale d'une course. Cependant, l'analyse comparative des performances des modèles inter-arrêts et du modèle chaîné montre qu'une amélioration des premiers n'améliore pas forcément le second.

## 1 Introduction

Dans le contexte actuel, il est d'une importance croissante de développer les transports en commun et d'en promouvoir l'utilisation. À cet effet, de nombreuses compagnies visent à prédire de manière efficace le temps de trajet de leurs véhicules, afin de mieux en planifier les trajets et identifier les sources d'éventuels retards. Cet article se place dans le contexte d'un réseau de lignes de bus, dans le but de simuler le comportement d'un bus au cours de son trajet. Dans cette optique, de nombreuses informations sont intéressantes. En effet, le temps de trajet total du bus de son départ à son terminus est important, mais le détail des événements survenant au cours de ce trajet l'est également, afin d'identifier les éventuelles causes de ralentissements et donc de retards du service et de modifier l'aménagement des lignes de bus en conséquence.

Nous étudions donc des modèles de régression visant non seulement à prédire la durée totale du trajet d'un bus, mais également la durée de ses différentes étapes. Pour cela, nous proposons d'utiliser un modèle prédictif distinct pour prédire le trajet du bus pour chaque inter-arrêt (i.e. entre deux arrêts successifs) d'une ligne, et chaîner les prédictions de ces modèles afin de prédire la durée de son trajet total. Enfin, nous proposons une nouvelle méthode d'entraînement visant à améliorer la performance de

modèles chaînés en modifiant les données d’entraînement afin d’y inclure le fait que le modèle recevra des données en partie prédites par le modèle précédent. Cette méthode, à notre connaissance, n’a pas encore été proposée dans l’état de l’art.

## 2 État de l’art

La prédiction de temps de parcours de bus a été explorée de diverses manières, y compris dans des travaux récents. Certains travaux utilisent des méthodes et modèles classiques d’apprentissage automatique tels que les machines à support vectoriel (SVM) (Bin et al., 2006 ; Yang et al., 2016), les K plus proches voisins (KNN) (Kumar et al., 2019 ; Liu et al., 2012) ou encore les arbres de décision de régression (RDT) (Cheng et al., 2018). D’autres travaux utilisent des méthodes plus poussées telles que les réseaux de neurones, pour des applications plus complexes, comme la prédiction de temps de parcours avec un modèle unique pour la totalité d’un réseau de bus (Shoman et al., 2020). Ces différents travaux s’accordent à dire qu’une des informations clés de la prédiction de temps de parcours est l’état de la circulation au moment du trajet. Or, Bin et al. (2006) soulignent qu’en l’absence de données en temps réel concernant la circulation routière, des informations sur le passé récent des trajets de bus sur la même ligne sont un bon indicateur de l’état de la circulation. C’est pourquoi certains chercheurs proposent de tenir compte des bus ayant récemment effectué le même trajet que le bus étudié. Par exemple, Petersen et al. (2019b) proposent d’utiliser des réseaux récurrents à mémoire court et long terme (LSTM) afin d’inclure des données sur les bus précédents ayant effectué le même trajet que le bus étudié. Petersen et al. (2019a) proposent également une méthode visant à décomposer le trajet en une séquence d’inter-arrêts afin de mieux prédire la durée de chacune d’entre elles et la durée totale de la course.

Pourtant, peu de travaux se sont intéressés aux relations entre les différentes étapes d’un même trajet d’un bus, préférant traiter un trajet comme un élément unique, prédit en une seule fois, ou en prédisant chacune de ses étapes indépendamment les unes des autres (Petersen et al., 2019a). Le fait de tenir compte des étapes précédentes lors d’une prédiction a pourtant été exploré dans plusieurs autres domaines. Par exemple, Sutskever et al. (2014) propose une formalisation de la prédiction d’un mot dans une phrase en tenant compte des mots qui le précèdent et qui ont été préalablement prédits. Plus proche de notre domaine d’application, Chen et Li (2019) propose de prédire l’éventuel retard d’un avion lors d’un trajet, puis d’utiliser cette prédiction pour prédire son éventuel retard sur les trajets suivants via une chaîne de modèles de régression. Cette même approche se retrouve dans la prédiction de gestes et de postures humaines proposée par Gkioxari et al. (2016). Chaque prédiction de posture est ici utilisée pour prédire les postures suivantes plus loin dans le temps.

## 3 Définitions et formulation du problème

Une ligne de bus est constituée de  $n$  arrêts notés par les entiers  $(1, 2, \dots, n)$ . Un bus voyageant le long de cette ligne passera par chacun des arrêts. Le trajet d’un bus

partant de l'arrêt de départ jusqu'au terminus d'une ligne est une course, notée  $x$ , avec  $x \in \mathcal{X}$ .  $\mathcal{X}$  est l'ensemble des courses effectuées sur la ligne. La durée d'une course  $x$  est notée  $\Delta$ . Le trajet d'un bus entre deux arrêts consécutifs  $i$  et  $i+1$  (aussi appelé l'inter-arrêt  $i$ ) lors de la course  $x$  est noté  $x_i$  et sa durée  $\Delta_i$ , où  $x_i \in \mathcal{X}_i$  avec  $\mathcal{X}_i$  l'ensemble des trajets entre les arrêts  $i$  et  $i+1$ . Une course  $x$  peut ainsi être décomposée en une séquence de trajets entre chaque inter-arrêt :  $x = [x_1, \dots, x_{n-1}]$ . La durée d'une course correspond donc à la somme des durées des inter-arrêts  $\Delta = \Delta_1 + \dots + \Delta_n$ .

Nous disposons d'un ensemble de variables décrivant les conditions au départ d'une course  $x$  ou d'un inter-arrêt  $x_i$ . Ce dernier inclut le retard au départ de  $x_i$ , cette donnée dépend donc de  $\Delta_{i-1}$ . Nous voulons obtenir un modèle capable de prédire à la fois la durée totale de la course  $\Delta$  ainsi que le temps de trajet de chaque inter-arrêt  $\Delta_i$ . Un modèle  $m$  est appelée modèle "de ligne" s'il produit une estimation  $m(x) = \bar{\Delta}$  de la durée d'une course  $x$ . De même, on appelle  $m_i$  un modèle "d'inter-arrêt" produisant l'estimation  $m_i(x_i) = \bar{\Delta}_i$ . Nous nous intéressons ici à des modèles dits chaînés permettant de produire une estimation de  $\Delta$  en combinant les résultats des modèles d'inter-arrêt  $(m_1, m_2, \dots, m_{n-1})$ . On cherche en particulier à déterminer si des modèles chaînés permettent d'égaliser des modèles de ligne en termes de qualité de prédiction.

## 4 Modèles chaînés de temps de parcours

Nous décrivons dans cette section deux modèles chaînés permettant de fournir à la fois des estimations des durées des trajets inter-arrêt et de la durée d'une course. Le modèle à prédiction chaînée correspond à une approche directe du problème, tandis que le modèle avec chaînage des données d'entraînement est l'alternative que nous proposons.

On peut prédire la durée d'une course en sommant des prédictions de durées de trajets sur des inter-arrêts consécutifs qui la composent. Il est donc possible de construire un modèle  $m_+$  de ligne à partir d'un ensemble de modèles d'inter-arrêts  $(m_1, \dots, m_{n-1})$ . Ce modèle chaîné de prédiction est noté  $m_+$ .

Le retard au départ de  $x_i$  étant une variable explicative de ce dernier et dépendant de  $\Delta_{i-1}$ , chaque  $m_i$  (excepté  $m_1$ ) reçoit des données construites à partir des prédictions du modèle précédent. Nous obtenons donc des trajets d'inter-arrêts modifiés  $\bar{x}_i$  où le retard au départ de l'arrêt  $i$  est le retard prédit par  $m_{i-1}$  sur  $\bar{x}_{i-1}$ . Grâce à ce procédé, nous effectuons une prédiction sur une course  $x$  comme suit :  $m_+(x) = m_1(x_1) + m_2(\bar{x}_2) + \dots + m_{n-1}(\bar{x}_{n-1})$ . Chaque étape de la prédiction tient compte des prédictions précédentes de proche en proche. Chen et Li (2019) utilisent, dans le cadre de trajets d'avions, un modèle unique pour prédire tous les trajets, tandis que nous utilisons un modèle pour chaque segment de course.

Lorsque le modèle  $m_i$  effectue une prédiction sur des données modifiées  $\bar{x}_i$ , il dispose d'un retard au départ de la course  $x_i$  qui a été prédit par  $m_{i-1}$  sur  $\bar{x}_{i-1}$  plutôt que le retard au départ observé dans nos données. Ce retard prédit comportera donc une erreur par rapport à l'observation qui va influencer la prédiction de  $m_i$ . Or, lorsque  $m_i$  est entraîné sur l'ensemble de données  $\mathcal{X}_i$ , les données qu'il reçoit en entrée sont bien celles observées. Afin de prendre en compte la dégradation de la qualité de ses

données en entrée, nous proposons d’entraîner  $m_i$  sur des données déjà prédites, et donc plus proches de ce qu’il recevra lors de son exploitation. Pour ce faire, pour chaque ensemble de données d’entraînement  $\mathcal{X}_i$  nous créons l’ensemble de données modifié  $\bar{\mathcal{X}}_i$ . Dans cet ensemble, les variables chaînées (*i.e.* retard du bus au départ de l’arrêt  $i$ ) sont remplacées en utilisant des prédictions précédentes  $\bar{\mathcal{X}}_{i-1}$ . L’ensemble de données  $\mathcal{X}_1$  n’ayant pas de prédécesseur,  $m_1$  n’utilisera pas de données prédites. Toutefois, l’approche directe consistant à construire itérativement  $\bar{\mathcal{X}}_i$  en utilisant les prédictions d’un modèle  $m_{i-1}$  construit sur l’ensemble  $\bar{\mathcal{X}}_{i-1}$  n’est pas robuste. En effet, prédire directement les valeurs d’un ensemble de courses sur lequel  $m_{i-1}$  a été entraîné amène un risque de sur-entraînement.

La méthode que nous proposons est donc la suivante. L’ensemble  $\bar{\mathcal{X}}_i$  est construit en différentes étapes par une méthode de type “*K-fold*” : pour chaque fold, un modèle  $m_{i-1}$  est entraîné sur  $K - 1$  parties de l’ensemble de données, puis effectue les prédictions de  $\bar{\Delta}_{i-1}$  pour la partie restante. Nous construisons itérativement chaque  $\bar{\mathcal{X}}_i$  en utilisant cette méthode. Une fois ces jeux de données créés, nous entraînons chaque  $m_i$  à nouveau sur les  $\bar{\mathcal{X}}_i$  finaux, donc sur l’ensemble des individus d’entraînement. Nous prenons ensuite la somme de chaque prédiction individuelle  $\Delta_i$  comme prédiction de  $\Delta$  à l’instar du modèle  $m_+$ . Nous appelons “modèle à entraînement chaîné” un modèle chaîné entraîné de cette manière et le notons  $m_+^K$ .

## 5 Description des expériences

Pour les expériences réalisées, nous considérons les données de 3 lignes de bus du réseau de transports publics de Marseille. Ces données sont constituées tout d’abord des données GTFS décrivant les horaires du réseau. À ces horaires théoriques s’ajoutent les données GPS recueillies par les compagnies de transport, indiquant l’heure de passage à chaque arrêt. Le système GPS étant sujet à des erreurs au départ et à l’arrivée des courses, nous omettons le départ et le terminus dans les trajets des bus.

L’ensemble de données  $\mathcal{X}_i$ , décrivant le parcours des bus sur une ligne entre les arrêts  $i$  et  $i + 1$  contient la date de la course, les informations météorologiques lorsque la course a été effectuée<sup>1</sup>, les heures de départ théorique du bus respectivement aux arrêts  $i$  et  $i + 1$ , ainsi son retard au départ de l’arrêt  $i$ . Pour finir, notre variable cible est le temps de trajet  $\Delta_i$  entre les arrêts  $i$  et  $i + 1$ . Notons que celui-ci inclut le temps d’attente à l’arrêt d’arrivée.

Afin d’assurer une bonne continuité entre les courses décrites dans  $\mathcal{X}$  et les différents  $\mathcal{X}_i$ , toute course comportant une erreur d’enregistrement pour un inter arrêt est enlevée dans son entièreté afin de ne conserver que les courses “complètes”. Enfin, chaque valeur aberrante de temps de trajet est remplacée par sa moyenne pour l’heure de départ concernée. Le tableau 1 donne les caractéristiques des lignes ainsi que le nombre de courses observées par ligne après nos pré-traitements.

Sur ces données, nous entraînons tout d’abord le modèle ligne appelé  $m_L$  sur chaque ligne. Ce modèle prédit directement la durée totale d’une course  $x$  en utilisant les

---

1. Canal officiel des stations météo française : [donneespubliques.meteofrance.fr](https://donneespubliques.meteofrance.fr)

Ligne	nb arrêts	nb course	temps moyen trajet (s)	Distance (m)
L18 Aller	28	16598	1695	7400
L18 Retour	27	24773	1782	7901
L4 Aller	26	3030	1449	7612
L4 Retour	24	2549	1360	7675
L89 Aller	26	2476	1781	5980
L89 Retour	22	2283	1703	5955

TAB. 1 – *Caractéristiques des lignes étudiées dans les expériences*

informations au début de la course (le deuxième arrêt dans notre cas). Il nous servira de base de comparaison. Nous entraînons ensuite deux modèles chaînés : le modèle chaîné classique  $m_+$  entraîné sur les ensembles de données  $\mathcal{X}_i$ , et le modèle chaîné  $m_+^K$  entraîné sur les ensembles de données  $\tilde{\mathcal{X}}_i$  pré-chaînés via la méthode k-fold.

Pour entraîner les modèles  $m_L$  ainsi que chacun des  $m_{i-1}$ , nous comparons trois algorithmes d’apprentissage automatique souvent utilisés dans l’état de l’art (Yang et al., 2016 ; Cheng et al., 2018 ; Kumar et al., 2019) : SVM, arbre de décision et KNN. Les hyper-paramètres de ces algorithmes ont été sélectionnés via une optimisation “gridsearch” pour chaque ligne et chaque inter-arrêt.<sup>2</sup>

## 6 Résultats

La figure 1 résume les résultats obtenus pour chaque modèle pour les lignes étudiées. En considérant les scores MAPE (*mean absolute percentage error*), on constate d’abord que les modèles chaînés ne sont pas systématiquement pires que le modèle  $m_L$ . On pourrait en conclure que la variable chaînée (*i.e.* le retard au départ de l’arrêt) n’a pas d’influence sur les prédictions des modèles. Or, une analyse des variables utilisées par les modèles (non rapportée ici) montre que c’est bien le cas : inclure le retard du bus au départ du trajet considéré augmente bien la précision des modèles  $m_i$  et  $m_L$ .

Notre proposition  $m_+^K$  présente des performances équivalentes ou supérieures à celles de  $m_+$  dans la grande majorité des cas : hormis pour la Ligne 89 Retour, les MAPE de  $m_+^K$  sont inférieures à celles de  $m_+$ . Il en est de même si on regarde les variations du coefficient de détermination  $R^2$ . Cela montre que l’entraînement chaîné utilisé pour créer  $m_+^K$  permet aux  $m_i$  de mieux prendre en compte la qualité dégradée de l’information apportée par le chaînage des prédictions, tout en continuant à l’utiliser. On peut toutefois noter que les résultats sont différents selon l’algorithme d’apprentissage utilisé. Ainsi, l’entraînement chaîné semble avoir un effet très limité pour RDT et beaucoup plus important pour les modèles construits avec SVR.

Afin d’améliorer la prédiction de la durée d’une course par un modèle chaîné, notre construction des  $\tilde{\mathcal{X}}_i$  pour  $m_+^K$  dégrade volontairement les données d’entraînement des modèles d’inter-arrêt  $m_i$ . On peut toutefois noter qu’une dégradation ou amélioration

<sup>2</sup>. Code et données des expériences : [https://gitlab.univ-nantes.fr/ferrettini-g/egc\\_experiment](https://gitlab.univ-nantes.fr/ferrettini-g/egc_experiment)

## Prédiction de temps de parcours de bus

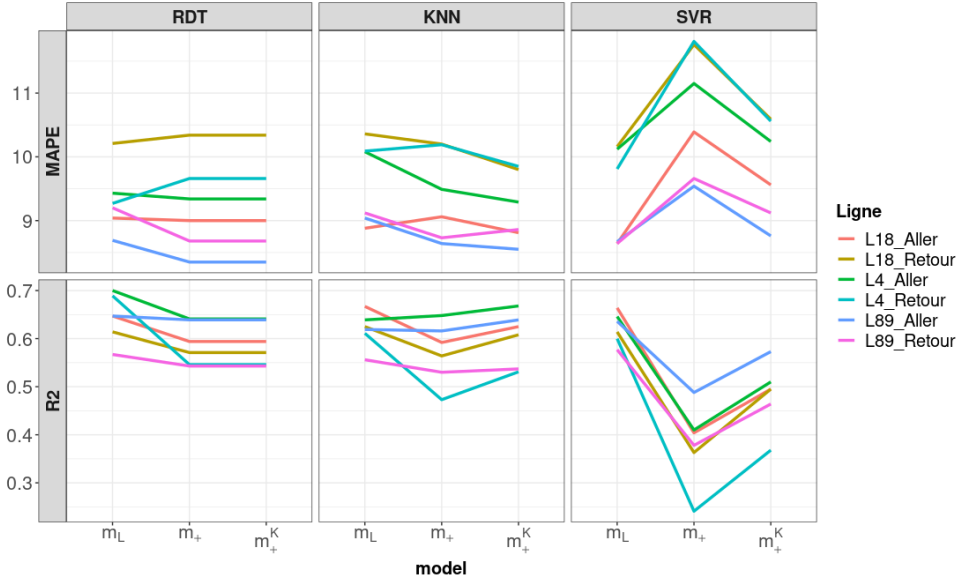


FIG. 1 – Comparaison des MAPE et des R2 pour les modèles  $m_L$ ,  $m_+$  et  $m_+^K$  pour la prédiction de la durée totale des courses.

moyenne des prédictions individuelles des  $\bar{\Delta}_i$  par les  $m_i$  ne correspond pas nécessairement à une variation réciproque de la qualité des modèles chaînés. En effet, il est possible que les erreurs individuelles des  $m_i$  se compensent pour aboutir à une prédiction plus précise des durées des courses.

Cet effet de compensation se retrouve expérimentalement si on compare à la fois la moyenne des erreurs des  $m_i$  et les erreurs des modèles chaînés selon l’algorithme d’apprentissage utilisé (voir la figure 2). On constate que les modèles SVR produisent des  $m_i$  avec une erreur moyenne dans tous les cas inférieure à celles obtenues avec les deux autres méthodes d’apprentissage. Or, comme noté dans la section précédente, les modèles chaînés construits avec SVR ont des erreurs supérieures lorsqu’ils prédisent une course entière. Cet effet est présent selon que l’on utilise notre construction par entraînement chaîné ou non.

Notons, par ailleurs, que notre méthode peut avoir un effet négatif notable sur l’erreur moyenne des modèles  $m_i$ . Cet effet est attendu par la façon dont les modèles sont entraînés dans notre construction. Cet accroissement de l’erreur moyenne des  $m_i$  est souvent moindre que la réduction de l’erreur des modèles chaînés. C’est particulièrement le cas pour les modèles utilisant SVR. Il existe toutefois des cas où on perd sur les deux tableaux (*e.g.* L89 Retour avec algorithme KNN).

La figure 2 indique qu’une amélioration ou dégradation des  $m_i$  ne correspond pas forcément à des meilleurs modèles chaînés. Dans notre cas, il faut choisir un compromis entre SVR et RDT selon que l’on souhaite avoir un modèle de ligne ou des modèles d’inter-arrêt plus précis (KNN est souvent inférieur à RDT sur les deux dimensions).

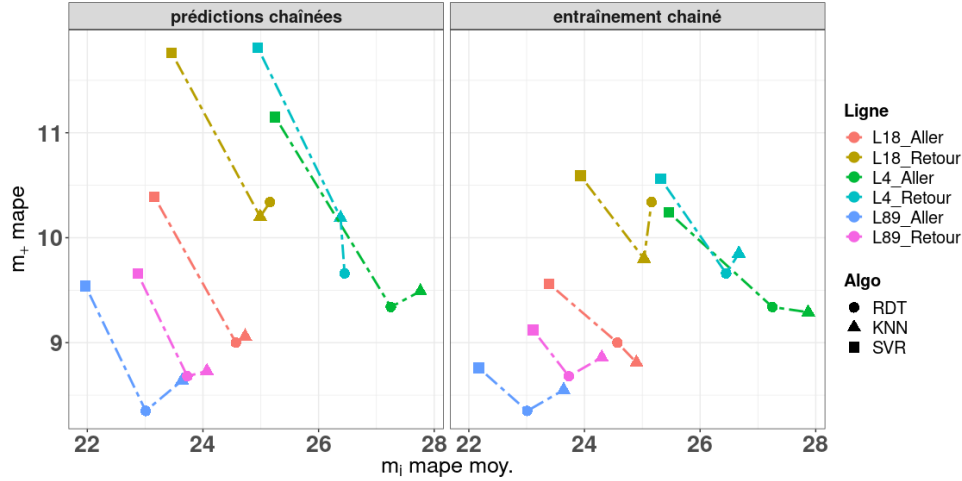


FIG. 2 – Comparaison des MAPE des modèles inter-arrêts  $m_i$  et des modèles chaînés  $m_+$  selon l’algorithme et l’entraînement utilisés.

Nous reviendrons dans la section 7 sur les possibilités d’optimisation multi-objectifs que ces observations appellent.

## 7 Conclusion et Perspectives

Dans le cadre de la prédiction de la durée d’une course de bus, nous avons pu voir que l’utilisation de modèles de prédictions chaînées produit des prédictions avec une précision inférieure. Ces derniers ont toutefois l’avantage de pouvoir également donner une prédiction des durées inter-arrêts. La transformation dans l’apprentissage des modèles chaînés que nous proposons permet d’améliorer la prédiction des durées des courses complètes. L’analyse de nos résultats suggère par ailleurs que l’amélioration individuelle des prédictions des temps inter-arrêts ne permet d’aboutir mécaniquement à une meilleure prédiction des temps de course complète.

Il serait pertinent d’explorer des compromis entre bonnes prédictions des inter-arrêts et des courses complètes. On aboutit alors à un problème d’optimisation multi-objectif *i.e.* les modèles chaînés acceptables formeraient un front de Pareto dans la figure 2. L’utilisateur pourrait en effet rejeter des modèles chaînés ayant une bonne qualité de prédiction des courses complètes, mais une mauvaise qualité de prédiction des durées inter-arrêts.

## Références

Bin, Y., Y. Zhongzhen, et Y. Baozhen (2006). Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems* 10(4), 151–158.

- Chen, J. et M. Li (2019). Chained predictions of flight delay using machine learning. In *AIAA Scitech 2019 forum*, pp. 1661.
- Cheng, J., G. Li, et X. Chen (2018). Research on travel time prediction model of freeway based on gradient boosting decision tree. *IEEE access* 7, 7466–7480.
- Gkioxari, G., A. Toshev, et N. Jaitly (2016). Chained predictions using convolutional neural networks. In *European Conference on Computer Vision*, pp. 728–743. Springer.
- Kumar, B. A., R. Jairam, S. S. Arkatkar, et L. Vanajakshi (2019). Real time bus travel time prediction using k-nn classifier. *Transportation Letters* 11(7), 362–372.
- Liu, T., J. Ma, W. Guan, Y. Song, et H. Niu (2012). Bus arrival time prediction based on the k-nearest neighbor method. In *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, pp. 480–483. IEEE.
- Petersen, N. C., F. Rodrigues, et F. C. Pereira (2019a). Multi-output bus travel time prediction with convolutional lstm neural network. *Expert Systems with Applications* 120, 426–435.
- Petersen, N. C., F. Rodrigues, et F. C. Pereira (2019b). Multi-output deep learning for bus arrival time predictions. *Transportation Research Procedia* 41, 138–145.
- Shoman, M., A. Aboah, et Y. Adu-Gyamfi (2020). Deep learning framework for predicting bus delays on multiple routes using heterogenous datasets. *Journal of Big Data Analytics in Transportation* 2(3), 275–290.
- Sutskever, I., O. Vinyals, et Q. V. Le (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27.
- Yang, M., C. Chen, L. Wang, X. Yan, et L. Zhou (2016). Bus arrival time prediction using support vector machine with genetic algorithm. *Neural Network World* 26(3), 205.

## Summary

The time required for a bus to finish its route and the inter-stops time are two relevant dimensions to predict in the context of bus route planning. If the full route is just the aggregation of the inter-stops, a solution is to train a chained model for each inter-stop and sum the individuals predictions to get the full route time. An efficient time prediction will use sequential information as input (*i.e.* the bus previous delay). However, model chaining fails to take into account the fact that each individual model use corrupted data (as prediction themselves) for its predictions while it was trained using observed data.

We introduced a new chained model that correct this issue by chaining not only the prediction but also the training step. Experiments show that our method improve full route prediction in most cases. Moreover, the comparative study of full route and inter-stops time predictions shows that improving the latter does not necessarily improve the former. These observations opens interesting perspectives involving multi-objective optimization procedures.