Noname manuscript No.
(will be inserted by the editor)

Impact of Data Cleansing for Urban Bus Commercial Speed Prediction

Gauthier LYAN · David GROSS-AMBLARD · Jean-Marc JEZEQUEL · Simon MALINOWSKI

Received: 26 April 2020 / Accepted: 4 March 2021

Abstract Public Transportation Information Systems (PTIS) are widely used for public bus services amongst cities in the world. These systems gather information about trips, bus stops, bus speeds, ridership, etc. This massive data is an inviting source of information for machine learning predictive tools. However, it most often suffers from quality deficiencies, due to multiple data sets with multiple structures, to different infrastructures using incompatible technologies, to human errors or hardware failures. In this paper, we consider the impact of data cleansing on a classical machine-learning task: predicting urban bus commercial speed. We show that simple, transport specific business and quality rules can drastically enhance data quality, whereas more sophisticated rules may offer little improvements despite a high computational cost.

Keywords prediction · machine learning · data cleansing · public transportation

1 Introduction

Current Public Transportation Information Systems (PTIS) produce a huge amount of heterogeneous data on a daily or even real-time basis. For example, smartcard systems, on-board bus units (AVLs[1]), schedule, referential, and real time radio bus monitoring systems are a gold mine to manage the bus network, useful to understand how it works and what to act on to improve it. For example, the average trip travel time, the schedule, the amount of km per year, the average bus stop spacing, etc. can be considered as inputs that, when properly processed, help the creation of service quality indicators for bus networks [7,6,5].

Many organizations are thus interested in trying to apply machine learning techniques to a wide part of their PTIS. These could be used to predict local and global behavior of the network, such as arrival time for each station [11], ridership [8] or even to elaborate what-if scenarios when considering road works or network enhancements.

However, the results of any prediction task may be inaccurate, due to omissions and errors in the input data set. PTIS are a typical example of error prone systems, for numerous reasons [10,12]. PTIS are made of many independent software and hardware that (try to) communicate using different, ad hoc protocols. Bus fleet are composed of hundreds of buses that embed numerous sensors and wireless communication systems that can fail. Risks of data loss come from the wireless communication system, the embedded hardware on buses and the number of buses in the fleet that have a variety of hardware to maintain. That variety increases the risk of errors, but also makes it difficult to identify the exact source of the error when it happens. Overall, cleansing this heterogeneous data is costly [13].

In this paper, we investigate the question of data cleansing for a typical machine learning task in PTIS: predicting bus commercial speed. We ground our analysis on the PTIS of Keolis Rennes that exploits the bus network of the city of Rennes, France, yielding 18GB of data per year.

In this specific domain, little is known [10,1] about the amount of effort one has to make to obtain accurate predictions. We consider a global data cleansing strategy with various levels of quality, ranging from easy/cheap

G. LYAN
E-mail: gauthier.lyan@keolis.com

D. GROSS-AMBLARD
E-mail: david.gross_amblard@irisa.fr

J.M JEZEQUEL
E-mail: jean-marc.jezequel@irisa.fr

S. MALINOWSKI
E-mail: simon.malinowski@irisa.fr

[1] Automatic Vehicle Location

ones to computational intensive ones, and observe their impact on prediction quality. We demonstrate experimentally that cleansing is mandatory, but also that a complete alignment/completion of all the available data sets, involving data synchronization and complex joins, is of little interest. More precisely, our contributions are the following:

- We define a datalake gathering all the information from the operation PTIS of Keolis, for a middle-size metropolis, Rennes, in France (730k inhabitants)
- We identify the errors in this datalake
- We present a dedicated data cleansing strategy capable of yielding different levels of data quality
- We evaluate the resulting commercial speed prediction precision on several, real size data sets.

The rest of this paper is organized as follows. In Section 2, we present the information system and the data sets we got from Keolis Rennes. In Section 3 we explain the various kinds of transformation and cleansing we can apply on the data. In Section 4 we compare the results of commercial speed predictions on two of our data sets with different level of data quality. In Section 5 we discuss the results and expose our threats to validity. Section 6 presents the related work before we conclude.

2 Data sources

Taking the Keolis Rennes example, we gathered data on a 6 months period between early July 2018 and late January 2019. The data collected are based on the bus network which contains 116 bus lines.

We had access to three data sets from the PTIS's AVL system of Keolis Rennes:

1. The referential and schedule: REF
2. The on-board central units data set: OCU
3. The radio real-time monitoring data set: RT.

The referential data set (REF) contains topological information over time. It describes the network as an oriented graph in which vertices are bus stops and edges are inter-points. An inter-point is a path between two bus stops. Bus lines are defined as paths through inter-points within the network graph. The schedule contains temporal information about the bus services, e.g scheduled time of arrival at bus stops over time. The referential and schedule volume of data per year is usually around 1 GB.

The on-board central units data set (OCU) contains *a posteriori* data. It is the record of bus trips structured as a table in which each row is a record of metrics of a bus serving an inter-point (or inter-stop), at a given instant. On-board sensors provide the data of this data set, hence it is expected to be accurate, but for reasons explained above, it is also incomplete and full of errors. It provides meta data such as bus, line, schedule information and metrics like commercial speed, travel time, traveled distance, etc. Most sensitive among these for contractual reasons, the commercial speed is the speed of the bus between two points, including travel time and dwell time at the origin point [7]. The OCU data set is made of 35 fields and represents around 10GB of data per year with daily file sizes varying between 9 to 40 MB.

Some of the missing data from the OCU data set however could be inferred from the radio real-time monitoring data set (that collects real time data by radio). Indeed, RT contains 13% more recordings than OCU.

The radio real-time data set (RT) is an historized data set. It is the record of real-time data traveling through TETRA[2]: it contains data that is somehow a duplicate of the data from the on-board units data set. However since it is a monitoring data set with a 20 seconds period, it only contains temporal information with little meta data, making it poorer, less precise and harder to use than OCU. Yet they are compatible with one another.

The RT data set has 20 fields and represent around 7GB of data per year with daily file sizes varying between 5 to 35 MB.

It is worth noting that both RT and OCU data sets contain commercial and deadhead trips. A bus trip is commercial when the bus is transporting passengers within a bus line service. Deadheads are network management trips that do not transport people, e.g trips between a deposit and the origin terminal of a bus line. This means that any kind of trip can be analyzed using these data sets.

3 Data Cleansing Strategy

In the OCU data set, readings for which bus speed was lower than 1 km/h or higher than the legal speed limit of 70 km/h, which respectively are low speed limit we defined and legal maximum speed for buses in France, can represent up to 5.4% of the data of OCU. Such information is absent from the RT data set that contains no metrics.

Still, considering the richer information available in OCU, it appears that it is the only production data set we have that can directly be used as a source for prediction tasks. To evaluate how much error correction is needed before prediction can be done efficiently, we ran 4 experiments, with the following data sets:

[2] Terrestrial Trunked Radio

- H0) raw OCU
- H1) Cleaned OCU, by applying business rules:
  - Illegitimate bus speed ($< 1km/h$ or $> 70km/h$) are deleted
  - null meta-data values that cannot be inferred are deleted
- H2) Inter-Points
  - Enriching OCU by merging with RT to complete missing timestamps for stop arrival and stop departure recordings
  - Joining with REF to update empty distances and add meta-data
  - Identifying and isolating trips to infer primary keys (trip departure, bus line, ...)
  - Recomputing commercial speed metrics with the new raw values

  Building this data set is costly because RT does have little metadata. Hence meta data such as starting hour of each trip, trip identification, etc must be inferred ”trip by trip”, making N computing tasks with N being the number of trips to recompute. **Table** 1 shows an overview of the process on a sample of data for a trip of the bus line 51. It shows that the base OCU data set is rich and ordered while base RT is poor and messy. Base RT needs to be ordered to extract individual trips and process them. Then the trips primary keys that include different fields such as trip departure information are rebuilt according to the minimum stop order within the trip, allowing the identification of missed trip starts (delayed or unplanned). Finally, the resulting enriched RT is merged with base OCU, keeping data from base OCU when it is valid.
- H3) Cleaned H2 using the same rules as H1.

H0 and H2 data sets are available here[3] in an anonymized and reduced version, in order to make the experiment reproducible.

4 Quality Experiments

4.1 Experimental setting

Table 2 shows the statistics and data quality variation of H0, H1, H2 and H3 data sets. It contains statistics about the bus network (i.e every inter-points in each data set) such as the total population of the data set (number of recordings), average commercial speed, commercial speed standard deviation, minimum and maximum commercial speed metrics in the data set (outliers), commercial speed error percentage (metrics that are under 1 km/h or over 70 km/h), number of identified inter-points in the data set and their average number of recordings.

Finally, this table contains the number of inter-points that are common to H0, H1, H2 and H3, that is the inter-points core we will use for the study.

Below is the global population variation. In accordance with **Table 2**, H2 and H3 are more populated than H0, and H1 and filtering a data set reduces the population of the resulting data set in all the cases.

- H0 → H1 : - 5,434 %
- H2 → H3 : - 2,616 %
- H1 → H3 : + 7,29 %

All of the data cleansing steps were done using a 4 cores 8 threads CPU @ 3.0 GHz with 32GB of ram @ 2933 MHz laptop running a 64 bits version of Ubuntu 19.04

- Preparing a sample of H0 or H1 data for a single day takes less than 5 minutes of computing and manual manipulations.
- Preparing a sample of H2 or H3 data for a single day takes around 2 hours of computing, involving a bottleneck of 1h+ for the RT preparation before merging with OCU.

4.2 Commercial speed prediction experiments

As set in the introduction, our stated goal is to measure the impact of data quality level on the prediction models’s precision.

For H0, H1, H2 and H3 we learn and predict bus commercial speed over 3119 common inter-stations given built-in features:

- Bus line ID
- Type of day
- Period in the day
- Month in the year
- Holidays time

[3] https://github.com/Tritbool/STAR_datasets

Table 1 H2 Creation process overview

Base OCU (lost tuples, missing values, erroneous measures)

line	time_start	direction	stop_id	previous_stop_id	arrival	departure	travel_time	distance	speed	...
?	?	?	?	?	?	?	?	?	?	...
?	?	?	?	?	?	?	?	?	?	...
51	17:34:00	A	1000	?	17:36:32	17:36:40	43	317	26.5	...
51	17:34:00	A	2346	1000	17:37:41	17:37:51	79	380	17.3	...
51	17:34:00	A	2347	2346	17:38:55	17:39:03	82	430	18.9	...
51	17:34:00	A	2356	2347	17:42:18	17:42:18	203	521	9.2	...
51	17:34:00	A	1981	2356	lost	lost	201	0	768.0	...
...	...	...	...	...	...	...	...	...	...	...

●

Base RT

line	direction	stop_id	arrival	departure	...
51	A	1000	17:36:32	17:36:40	...
3	R	3452	11:21:22	11:21:25	...
51	A	2347	17:38:55	17:39:02	...
12	A	1234	13:21:02	13:21:17	...
51	A	4021	17:32:00	17:34:00	...
1	R	1000	8:02:00	8:02:21	...
51	A	1001	17:35:07	17:35:18	...
1	A	1287	8:03:00	8:03:01	...
51	A	2356	17:42:18	17:42:18	...
51	A	2346	17:37:41	17:37:55	...
51	A	1981	17:42:52	17:43:01	...
...	...	...	...	...	...

⇓

Enriched RT (requires to locate & synchronize trips using REF database)

line	time_start	direction	stop_id	order	arrival	departure	travel_time	distance	...
1	8:02:00	A	3657	1	8:02:00	8:02:21	0	0	...
1	8:03:00	R	1287	4	8:03:00	8:03:01	-1	457	...
3	11:00:00	R	3452	16	11:21:22	11:21:25	77	298	...
51	17:34:00	A	4021	1	17:32:00	17:34:00	0	0	...
51	17:34:00	A	1001	2	17:35:07	17:35:18	78	316	...
51	17:34:00	A	1000	3	17:36:32	17:36:40	43	317	...
51	17:34:00	A	2346	4	17:37:41	17:37:55	49	267	...
51	17:34:00	A	2347	5	17:38:55	17:39:02	69	430	...
51	17:34:00	A	2356	6	17:42:18	17:42:18	67	521	...
51	17:34:00	A	1981	7	17:42:52	17:43:01	43	282	...
...	...	...	...	...	...	...	...	...	...

⇓

Base OCU × **Enriched RT**

line	time_start	direction	stop_id	order	arrival	departure	travel_time	distance	speed	...
...	...	...	...	...	...	...	...	...	...	...
51	17:34:00	A	4021	1	17:32:00	17:34:00	0	0	-1.0	...
51	17:34:00	A	1001	2	17:35:07	17:35:18	78	316	14.6	...
51	17:34:00	A	1000	3	17:36:32	17:36:40	43	317	26.5	...
51	17:34:00	A	2346	4	17:37:41	17:37:51	79	380	17.3	...
51	17:34:00	A	2347	5	17:38:55	17:39:03	82	430	18.9	...
51	17:34:00	A	2356	6	17:42:18	17:42:18	203	521	9.2	...
51	17:34:00	A	1981	7	17:42:52	17:43:01	43	282	23.6	...
...	...	...	...	...	...	...	...	...	...	...

All the features were selected knowing that the most important one we have built-in is the period in day[9], then normalized and missing values were imputed using K-Nearest Neighbours imputation, with K set to 10. Thus, the missing data is imputed using the data of the 10 nearest neighbours of the erroneous row, according to which class of neighbours is more numerous within the bench of neighbours.

We learned the speed on the different inter-points of the network in such a manner that the output is the predicted bus speed for a given bus line at a given period of the day on a given and already known inter-point.

Table 2 data sets properties

data set	Population	Average speed	Speed standard deviation	Minimum speed	Maximum speed	Speed error rate
H0	16793293	20.31 km/h	11.6 km/h	-996 km/h	130 km/h	5.4%
H1	15880770	21.33 km/h	9.76 km/h	1 km/h	70 km/h	0%
H2	17496142	20.77 km/h	10.3 km/h	0 km/h	70 km/h	2.6%
H3	17038432	21.33 km/h	9.9 km/h	1.1 km/h	70 km/h	0%

data set	Number of Inter-Points	Average Population per inter-point
H0	10262	1636
H1	8398	5041
H2	3363	5234
H3	3329	5149

Total H0,H1,H2 and H3 common inter-points
3119

Hence, we predict the speed using only temporal and categorical built-in features, ignoring the spatial information[4] knowing that inter-points are only 607 meters long in average, begin and end with a bus stop.

Using the data sets presented before, we trained a set of 6 different prediction models from the Scala SMILE framework[5] in its 1.5.2 version:

1. Random Forest
2. Decision Tree
3. Lasso
4. Bayesian Ridge
5. Gradient Boosting
6. Ordinary Least Square

We chose these regression models because they are common and long known as well as for their ease of use and overall performance.

We used 10-fold cross validation on each model to configure the models's hyper-parameters and kept the best resulting prediction's RMSE amongst them.

The Root Mean Squared Error is the standard deviation of the predictions's errors :

$$\sqrt{\frac{1}{N}\sum_{1}^{N}(Y_t - \hat{Y_t})^2} \tag{1}$$

N is the population of the test data set, Y_t the observed commercial speed and $\hat{Y_t}$ the predicted commercial speed. A low RMSE (i.e near 0) indicates a good prediction accuracy.

The RMSEs results are compared in Figures 1, 2, 3, 4 and 5.

Figures 1 to 4 compare the resulting RMSE for different pairs of data sets, respectively H0-H1, H2-H3, H0-H2 and H1-H3. On x axis are presented the inter-points ordered by name (for a better readability of the legend, only a few names are displayed over the 3119 inter-points). On y axis are presented the RMSE values for each inter-point.

Figure 5 summarizes the RMSEs of H0, H1, H2 and H3 in order to facilitate the analysis of results

Also we compared RMSEs normalized on scatter indicator of input data (here standard deviation, and mean) in Tables 4 and 5. The nearer to 0 it is, the better the fit is. De facto, it is harder to get a normalized RMSE close to 0 normalizing with standard deviation than with mean.

The prediction models were trained on a 64 cores @ 3.4Ghz machine with 64 Gb RAM @ 2933Mhz, running Manjaro 18.1 in its KDE version.

Computation time for the 3119 inter-points of each data set is around 14 hours and detailed as follows:

- 3h30 data preparation
- 10h30 of training for 3119 inter-points * 6 models

During prediction experiments, some inter-points'training would fail, either because of data issues (like insufficient amount of data, malformed data), or tweaking issues (as we tweaked each model only once in accordance to the average statistics of the inter-points). **Table 3** shows the failing statistics for each input data set.

Figure 5 shows the variation of prediction accuracy between the different data sets.

Table 3 Inter-points learning failures

data set	Amount of failures
H0	160
H1	160
H2	126
H3	186

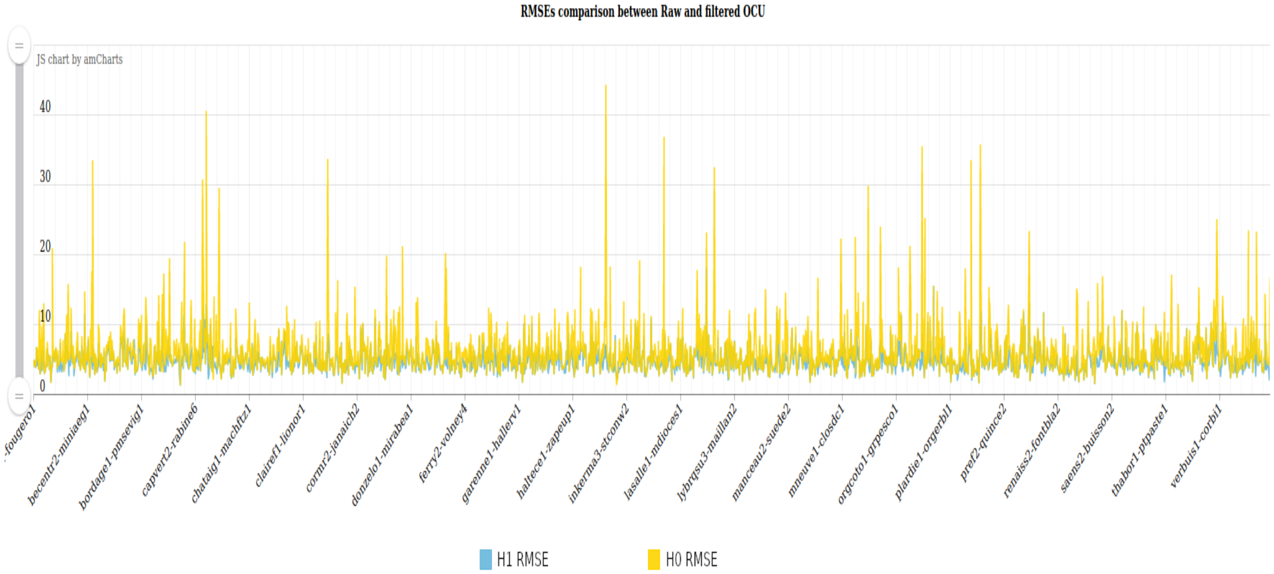

Fig. 1 RMSE variation over 3119 inter-points common to H0 and H1

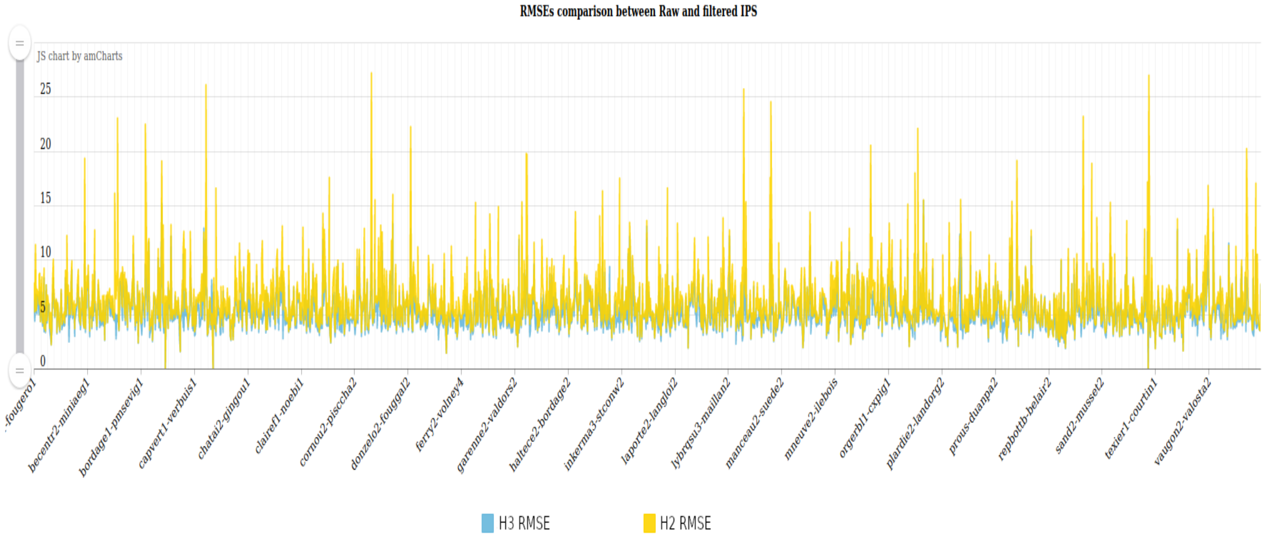

Fig. 2 RMSE variation over 3119 inter-points common to H2 and H3

5 Discussion

Figure 1 shows the RMSESs of the inter-points of H0 and H1. The content of **Figure 5** supports the fact that the quality enhancement done from H0 to obtain H1 implies a statistically significant boost in predictions with the average RMSE of predicted H1 being 21.34% smaller than predicted H0. In the same way, the 95th percentile in predicted H1 is 33.09% smaller than the 95th percentile of H0. In the same way, the standard deviation of the RMSE of H1 benefits from a 57.87% drop in comparison to the standard deviation of the RMSE of H0..

[4] That we don't have anyway

[5] http://haifengl.github.io/

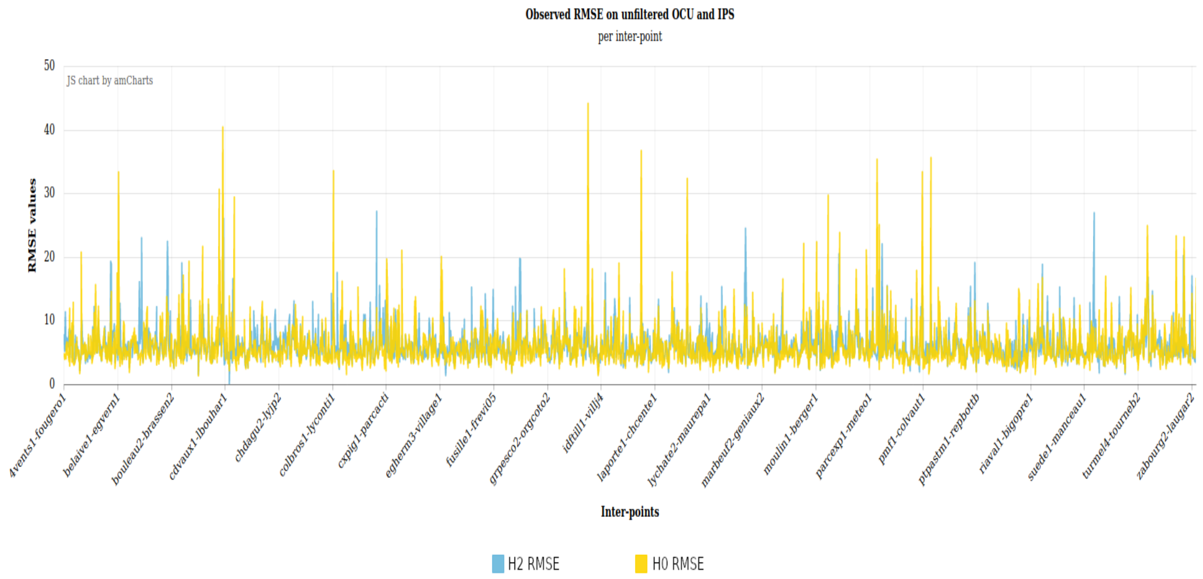

Fig. 3 RMSE variation over 3119 inter-points common to H0 and H2

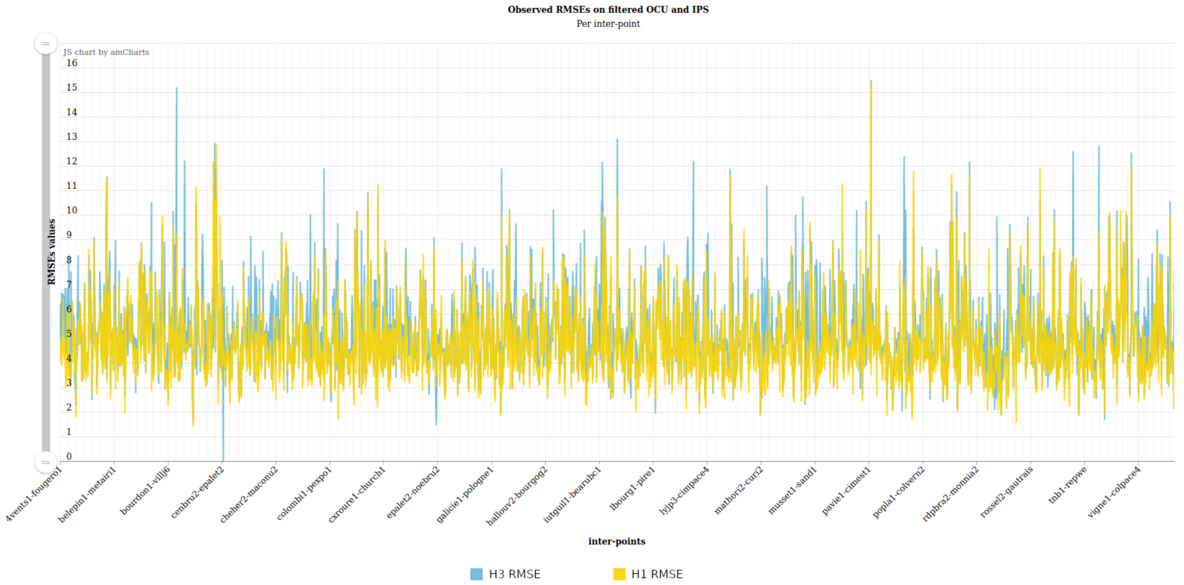

Fig. 4 RMSE variation over 3119 inter-points common to H1 and H3

Table 4 NRMSE of H1

SD normalized RMSE	Mean normalized RMSE
0.84	0.2

Table 5 NRMSE of H3

SD normalized RMSE	Mean normalized RMSE
0.91	0.23

In **Table** 2, the filtering of H0, yielding H1, shows that the average population per inter-point dramatically raises, while the number of represented inter-points drops by 18.16%. We note that the global population varies in the exact same proportion as the input data error rate. This is due to the fact that the amount of faulty data in the global population is equal to the error rate of the global population.

Figure 3 shows the RMSESs of the inter-points of H0 and H2. **Figure 5** contains insights that let us think that the quality enhancement done from H0 to obtain H2 does not really enhance the prediction precision. Indeed, with an increase of 1.14% of the average RMSE from H0 to H2, it seems to be of little interest when contrasted

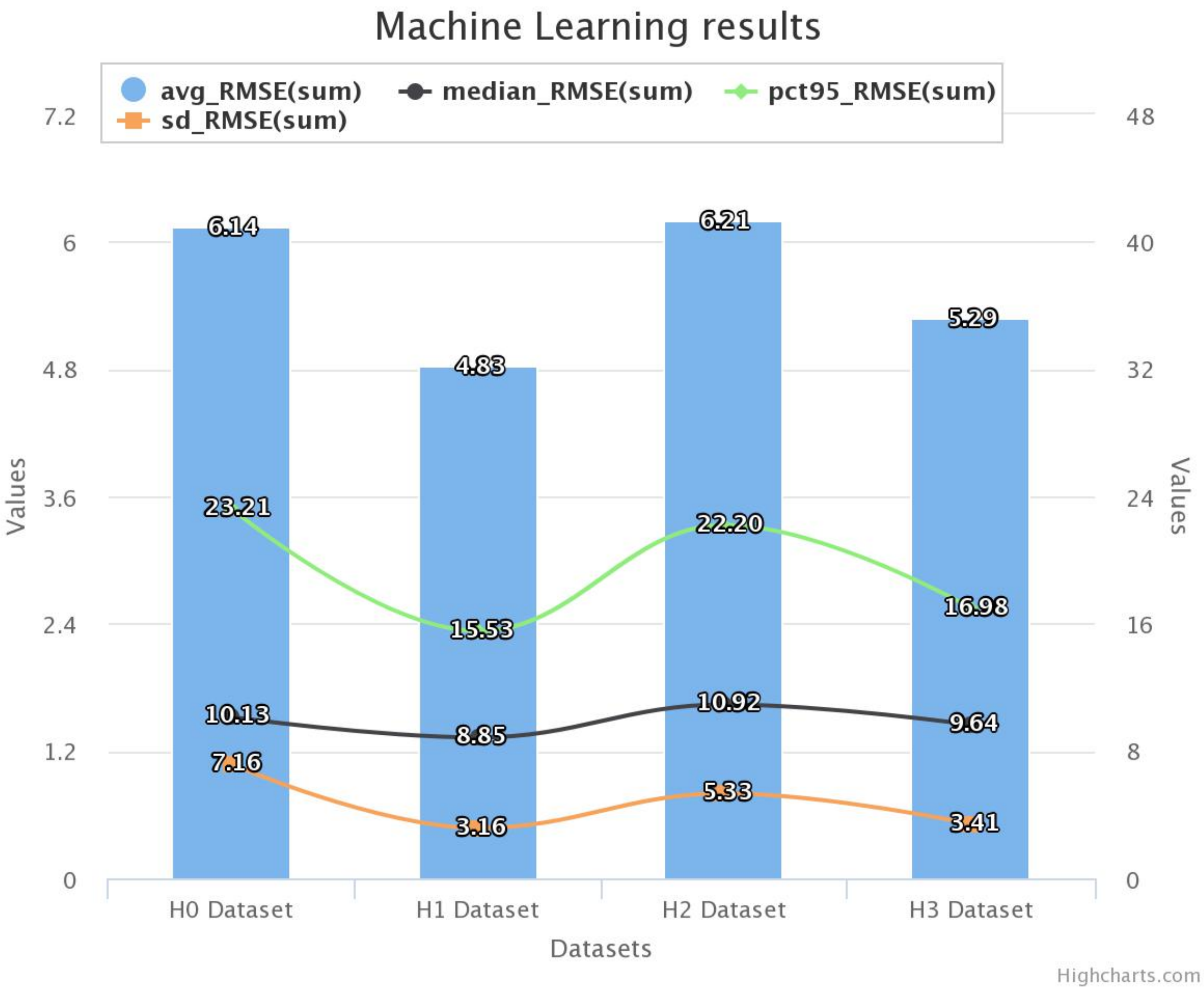

Fig. 5 Prediction results comparison chart

with the computational time needed to process a single day of data for H2. Indeed, processing a day of data for H2 is 24 times longer than computing a day of data for H0.

However, as shown in **Table** 2, the speed error rate is lowered by around 51,85% from H0 to H2. In the same way, H0 outliers are far more distant from the mean than H2's. On the other hand, it is worth noting that even if the average inter-point population is more than doubled from H0 to H2, the standard deviation of the commercial speed only varies by around 10%. Also, the mean speed is quite steady between those two data sets. We observe that the average population per inter-point of H1 and H2 is not much different with only 2% of variation. Added to the heavy computing needed to build H2 from H0, as stated before, those results make it quite obvious that the cleansing from H0 to H2 is somehow an overkill compared to the cleansing from H0 to H1 in terms of data set statistics and prediction performance.

Figure 5 confirms that H2 is somehow not suitable for commercial speed prediction, as stated before. Also, **Figure** 5 highlights the evidence that the cleansing from H2 to H3 is worthy, filtering H2 being less costly in computing time than building it. Filtering a day of data from H2 yielding H3 takes a few minutes. The global population varies in the exact same proportion as the error rate, following the same rules and trends of the population evolution between H0 and H1.

Finally, **Figure** 5 shows that even if H3 is more populated than H1 (cf. Table 2), the enhancement of the standard deviation and 95th percentile of RMSEs are not sufficient to make it worth the cost to compute H2 and H3 for prediction. Worse, the average RMSE's of H3 increases even if the average speed and standard deviation of H1 and H3 are similar as stated in Table 2, and the normalized RMSEs in **Tables** 4 and 5 supports the idea that H1 is the best choice for prediction in this specific context.

5.1 Threats to validity

5.1.1 Construct

In the context of this paper, construct validity implies that the data chosen for the experiment is representative enough regarding the original data and the data quality rules created on purpose. Also the prediction models should behave evenly when an experiment is led several times using the same datasets.

Prediction models were not tuned using grid search because of time issues and compatibility in the SMILE API. This implies that the results could be better with enhanced hyper-parameters, yielding less training failures. In our investigation, we found that many of the inter-points for which prediction failed were poor in data. Also, the amount of failures in our study represent less than 6.5% of the 3119 inter-points we had in average. Recovering all of the failures could change the overall result, yet, probably not dramatically even with a better tweaking of hyper parameters.

We assumed that the inter-points were small enough not to consider the spatial information. May be it would be interesting to confirm whether this was right or not in our further work. Usually [7], the ratio of elements (such as traffic lights, stops, etc) per kilometer is considered to estimate their impact on bus commercial speed. Our inter-points being smaller than 1 km, they probably are short enough for those elements not to impact the commercial speed significantly. However, adding the spatial data to our data set in further studies is needed to confirm this claim.

Inter-points that are exclusive to each data set were not tested and may have changed the interest of H2 and H3, hence the global result of the study. If one wants to learn and predict commercial speed over the whole network (commercial and deadhead trips), then H0 would be the way to go according to the fact that it contains far more inter-points information. However, if one wants to work on commercial trips only, H1 or H3 would be better. Hence, depending on the goal, the outcome might be different. In our case, we aimed at predicting the commercial speed regardless of the commercial or deadhead aspect of the input trips data.

5.1.2 Internal threats to validity

We selected features among dozens of fields based on what others studies have used to predict commercial speed. It is unlikely that other features that the one we used would have significantly enhanced the prediction results, yet it is still a threat to consider.

We assumed that the 20 seconds temporal resolution of the data in RT has no specific impact on the bus travel time, the amount of trips smoothing it out. We somehow validate that there is indeed a smoothing effect when we compare the average speed of H1 and H3 in Table 2 which are equal. However the standard deviation evolution shows that the 20 second resolution has an impact, even if it is quite low. If we had to recover more than 13% of data from RT, may be the resolution would have a greater impact unless the whole data set was recomputed from RT only.

We did not control the data quality of timestamped data and odometer distances metrics. More generally, we have no mean to assess the accuracy level of the AVL that provides us with this data. Even if we cannot control the data upstream, we observe that the data consistency is quite good: as already discussed, H0 only has 5.4% of errors (cf Table 2). If the odometer were to fail massively, we would probably observe a rise in speed error rate, making it clear that we should not use this data as an input for our experiments.

5.1.3 External threats to validity

Cleansing rules were chosen based on what is known about legal bus speeds in France and it is possible that the filtering could be better done. Typically, each inter-point would probably be predicted more accurately if they were to have their data filtered against their own specific legal maximum speed, which was not possible to consider in this study. Based on the fact that bus driver are urged to respect speed limits in the different areas the bus travel through and that this is a professional condition, it is probably legitimate to assume that the speed recorded within the different inter-points stays within the legal limits in almost all of the cases. Yet a significant rise of the commercial speed would be an indicator of driving issues.

This study was based on the bus data of a single city, Rennes (France), for a single period of time. We mitigated the single period of time issue by running the same experiment in varying periods, without noticeable impact on our results. The Rennes metropolis is typical of Western European cities, with a crowded medieval downtown, less crowded suburbs, and some reserved ways for public transportation, so we foresee that most of what we have learnt here could be applied to similar cities. The same goes for the data quality issues: as we have seen before, most of Public Transportation Information Systems suffer from similar data quality issues. Still before being able to generalize our results, it would be necessary to run similar experiments with other cities bus networks. This could be easy to do since our data processing methods could be reused out of the box for a new data set.

6 Related work

6.1 Transport related

There are only a few studies on data quality in transportation.

Brian L. Smith et al. [1] proposed a set of statistical tools to handle missing data in transportation management systems. Their work shows that it is possible to infer missing data with a satisfying error rate, as long as the correct heuristic is used for this purpose. Their work is part of the world of global transport while ours is part of the world of bus transport. Also, we rely more on filtering and redundancy than on empty data imputation to enhance the quality of our data sets.

Ma and Chen [10] explored the data quality of smart card and GPS systems. They claim that it is needed to use redundant data when it is available to recover or correct missing values and that any useless field should be removed from the smart card data sets. Also, they claim that the error identification is made harder by the complexity of transportation systems that are often made of many manufacturers and stakeholders. They say that this should urge the data manager to work on the data consistency before data collection to make sure that all the production data is based on the same units, definitions and same accuracy level. We took their advice when building H2 and H3 data sets, especially regarding the use of redundant data. However, we applied it on AVL data set instead of smartcard. Also, we managed to obtain a satisfying data quality level after the data collection was made without knowing how the data is managed upflow.

Robinson et al. [12] studied methods to improve data quality of smartcard systems. They identify different sources of error in 4 investigation domains that are software, hardware, data and user. They proposed a method to identify boarding and unboarding data errors and faulty data supplies. They claim that taking care of smartcard data quality can reduce costs and enhance the service quality offered by the transportation network. While their work focused on smartcard data, we focused on AVL data, taking the errors sources they found in AVL systems into account, especially odometer issues. This led us to use theoretical distances when needed and to recompute commercial speed metrics in H2. Taking their advice into account led us to the conclusion that the AVL system in Rennes looks quite accurate, even if some filtering is needed after data collection.

Also, there is a limited number of studies on the impact of data quality on prediction accuracy.

6.2 Other domains

Cortes et al. [4] studied the impact on data quality for classification tasks. They used a data set from AT&T to predict failures on telecommunication paths. They ran series of experiments using neural networks to reach an asymptotic value of 25% classification error which is yielded by the input data quality. These results made them claim that data quality is paramount for classification tasks. More generally, they say that using a data set that was not thought for a specific task at hand may yield poor results. They focused on neural network approach while we used a set of different prediction (machine learning) algorithms, yet we obtained results that follow the same trend as described in their study with a prediction accuracy reaching a floor at a given level of data quality.

Blake and Mangiameli [3] worked on data quality and problem complexity (number of categories to classify) impact on classification tasks. Their study is composed of four dimensions: accuracy, completeness, consistency, and timeliness that are considered to be the most important dimensions in data quality. They imagined 4 hypothesis based on the idea that the variation of the different data quality dimensions and the problem complexity imply classification accuracy variations. They first assessed their hypothesis over a home-made tool based on Weka[6] using generated data as input. Finally they validated their hypothesis on a real-world use case provided by Tel-Fast. While they used classification algorithms, we worked on a set of regression algorithms. Also, we did not explore the timeliness dimension of our data set, nor the complexity of the problem to solve, because we only had one. Yet, our results are validating the fact that, in our context, accuracy and consistency have an impact on prediction while completeness does not have a significant impact.

Bansal et al. [2] demonstrated that there is a correlation between data accuracy and prediction level for linear regression and neural networks models on financial risk forecasts. They used data from the Capital Markets Sector of Manufacturers Hanover Trust Company and CITIBASE, in which they added noise at different controlled level to assess the impact of data quality level on their machine learning tasks. Their results clearly show that the more the data is noised, the worst the results in most of the cases for linear regression and neural networks. Their study focused on linear regression and neural network while we use a set of prediction (machine learning) algorithms. Also, we tried to lessen the noise in our data while they added some in theirs. Finally, they claim that in most of the cases the better the data quality is, the better the prediction will be. Our work partly validates this claim, yet at some point the data quality process becomes too costly in our context with regard to the gain of prediction accuracy, that starts to stagnate.

[6] https://www.cs.waikato.ac.nz/ ml/weka/

7 Conclusion

In this paper we worked on 4 different data sets that gather 6 months of data from the Public Transportation Information System of the bus network of Rennes, France. We used different data quality enhancing techniques and compared the resulting statistics, and prediction usability of those data sets. It appeared that, in this context, data quality is needed to ensure acceptable data sets statistics and prediction accuracy. However, over qualifying the data implies a high ratio of computational cost against performance gain, making it unadvised for non-production purposes like prediction experiments. Finally, one would have to define a threshold for the ratio between computation cost of data quality enhancement versus gain in prediction accuracy over which it is considered counter productive to enhance more the data quality of the input data set.

8 Compliance with Ethical Standards

- **Conflict of Interest**: All of the authors declare that they have no conflict of interest.
- **Funding**: Not applicable to this study.
- **Ethical approval**: This article does not contain any studies with human participants or animals performed by any of the authors.
- **Availability of data and material**: Data used for this study will be freely available at the publication.

References

1. B. Smith, W.S., Conklin, J.: Exploring imputation techniques for missing data in transportation management systems. Transportation Research Record: Journal of the Transportation Research Board, vol. 1836, pp. 132–142 (2003)
2. Bansal, A., Kauffman, R.J., Weitz, R.R.: Comparing the modeling performance of regression and neural networks as data quality varies: A business value approach. Journal of Management Information Systems **10**(1), 11–32 (1993). DOI 10.1080/07421222.1993.11517988
3. Blake, R., Mangiameli, P.: The effects and interactions of data quality and problem complexity on classification. Journal of Data and Information Quality **2**(2), 1–28 (2011). DOI 10.1145/1891879.1891881
4. C. Cortes, L.D.J., Chiang, W.P.: Limits on learning machine accuracy imposed by data quality. Proceedings of the First International Conference on Knowledge Discovery and Data Mining (1995)
5. C. E. Cortés J. Gibson, A.G.M.M., Zúñiga, M.: Commercial bus speed diagnosis based on gps-monitored data. Transportation Research Part C: Emerging Technologies (2011)
6. Courtois, X., Dobruszkes, F.: L'(in)efficacite des trams et bus à bruxelles, une analyse désagrégée. The e-journal for academic research on Brussels (2008)
7. Fernandez, R., Valenzuela, E.: A model to predict bus commercial speed. Traffic Engineering & Control, vol. 44 (2003)
8. J. Deng, H.N., Chen, C.: Research on bus passenger traffic forecasting model based on gps and ic card data. in Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019) (2019)
9. Kotagiri, Y., Pulugurtha, S.S.: Modeling bus travel delay and travel time for improved arrival prediction. In: International Conference on Transportation and Development 2016. American Society of Civil Engineers (2016). DOI 10.1061/9780784479926.052
10. Ma, X., Chen, X.: Public transportation big data mining and analysis. In: in Data-Driven Solutions to Transportation Problems. Elsevier (2019)
11. Mehmet Altinkaya, M.Z.: Urban bus arrival time prediction: A review of computational models. International Journal of Recent Technology and Engineering (IJRTE) (2013)
12. Robinson, S., Narayanan, B., Toh, N., Pereira, F.: Methods for pre-processing smartcard data to improve data quality. Transportation Research Part C: Emerging Technologies **49**, 43–58 (2014). DOI 10.1016/j.trc.2014.10.006
13. Wang, L.: Heterogeneous Data and Big Data Analytics. Automatic Control and Information Sciences p. 8 (2017)