Anomaly detection in smart card logs and distant evaluation with Twitter: a robust framework

E. Tonnelier[a,*], N.Baskiotis[a], V.Guigue[a], P.Gallinari[a]

[a] *UPMC - Sorbonne Universités - LIP6 - CNRS, 4 place Jussieu, 75005 Paris*

Abstract

Smart card logs constitute a valuable source of information to model a public transportation network and characterize normal or abnormal events; however, this source of data is associated to a high level of noise and missing data, thus, it requires robust analysis tools. First, we define an anomaly as any perturbation in the transportation network with respect to a typical day: temporary interruption, intermittent habit shifts, closed stations, unusual high/low number of entrances in a station. The Parisian metro network with 300 stations and millions of daily trips is considered as a case study. In this paper, we present four approaches for the task of anomaly detection in a transportation network using smart card logs. The first three approaches involve the inference of a daily temporal prototype of each metro station and the use of a distance denoting the compatibility of a particular day and its inferred prototype. We introduce two simple and strong baselines relying on a differential modeling between stations and prototypes in the raw-log space. We implemented a raw version (sensitive to volume change) as well as a normalized version (sensitive to behavior changes). The third approach is an original matrix factorization algorithm that computes a dictionary of typical behaviors shared across stations and the corresponding weights allowing the reconstruction of denoised station profiles. We propose to measure the distance between stations and prototypes directly in the latent space. The main advantage resides in its compactness allowing to describe each station profile and the inherent variability within a few parameters. The last approach is a user-based model in which abnormal behaviors are first detected for each user at the log level and then aggregated spatially and temporally; as a consequence, this approach is heavier and requires to follow users, at the opposite of the previous ones that operate on anonymous log data. On top of that, our contribution regards the evaluation framework: we listed particular days but we also mined RATP[1] Twitter account to obtain (partial) ground truth information about operating incidents. Experiments show that matrix factorization is very robust in various situations while the last user-based model is particularly efficient to detect small incidents reported in the twitter dataset.

1. Introduction

Transportation networks have become a crucial urbanization planning tool: in dense urban areas, most people rely only on public transportation system to move, to go to work, to visit friends or for entertainment trips. Understanding, predicting and characterizing transportation network failures is critical to improve the whole system and provide a reliable service. It has been shown that a good transportation system increases public health, boosts the economy, saves space and time inside a city [1]. Decision makers have to rely on strong indicators to pursue coherent development policies. Until last decade, expert knowledge and population surveys were the usual and most accurate techniques to apprehend the behavior of a transportation network [2]. Smart cards revolutionized the field with the opportunity to obtain massive accurate data on the user's mobility and, thus, to analyze the use of a transportation network. Research exploiting this data has been conducted on numerous applications, such as detection of meteorological events [3],

[1] Parisian transport authority

[*] Corresponding Author

Email addresses: `emeric.tonnelier@lip6.fr` (E. Tonnelier), `nicolas.baskiotis@lip6.fr` (N.Baskiotis), `vincent.guigue@lip6.fr` (V.Guigue), `patrick.gallinari@lip6.fr` (P.Gallinari)

prediction of congestion [4], characterization of users habits [5] and prediction of individual trips [6]. Exploiting log flows enables to catch habits on a mid/long-term basis, it provides strong dynamic mobility flow information and it gives a new view on service quality and customer satisfaction [7]. However, log flows often contain a high level of noise and missing data[2], they occur on a complex graph structure and have multi-scale aspects (the density of logs highly depends on the time of the day and the overall number of logs varies from one station to another).

This article focuses on the task of anomaly detection on smart card logs in an unsupervised setting, i.e. without having any knowledge on the time periods of anomalies. Anomaly detection is a widely studied topic in several application domains [8], for instances in computer security [9], fraud detection [10], . . . From a machine learning perspective, there are two main contexts depending on the availability of labels indicating the anomalies. If labels are available, usual supervised machine learning algorithms are relevant to extract meaningful patterns from the data (Neural Networks [11], Shapelets-type [12]). When no supervision is available -like in our application-, methods are generally based on clustering algorithms to identify outliers (K-Means [13], Density-based clustering [14], KNN [15]).

We define an anomaly as a spatiotemporal event -attached to a given station and during a certain time window-corresponding to the deviation of the station activity from its regular model. We focus on short anomalies occurring inside a usual day; thus, the proposed approaches will use daily reference models describing the station behavior during 24 hours; we simply separate the seven days of the week for every station. According to [3], the weekly regularity hypothesis is relevant at the season scale and we mainly work on a 3 months dataset. In this study, four different approaches are explored to detect network anomalies. The first three proposals rely on averaged models for every day-station; indeed, log flows are very noisy and aggregating several samples corresponding to a same phenomenon is a standard noise reduction strategy. The first two proposed approaches naively use those daily averaged models as prototypes for each couple of station and day of the week. Anomalies are detected by using a threashold on the L_1 distance between a given raw station daily log and the corresponding prototype. Two variants are considered: the first one using the raw number of check-in while the second one using the normalized signal, less subject to volumetric differences. Those two first approaches can be seen as a strong kernel method which tackle anomaly detection as outliers identification ([16]). The third approach relies on a Non-Negative Matrix Factorization (NMF) algorithm to learn a denoised and compact representation of the daily temporal station profiles. NMF algorithms learn simultaneously the atom dictionary and the parsimonious weights. Both are used to reconstruct the original signal, thus, encouraging the emergence of atoms shared across a number of profiles and avoiding overfitting. Our contribution consists in an NMF improvement that enforces time consistency during decomposition and increases the robustness of the latent representation. Then, an anomaly detection procedure operating directly in the latent space is proposed; both (station, day) couples and references are mapped in the latent space and compared using a L_1 distance. The fourth proposed approach is based on user modeling. Anomalies are measured for each log based on an individual spatial model. The problem is turned into an anomaly aggregation problem: detecting real events in an anomaly flow. We perform a temporal convolution on the station graph that tackles this issue. This heavier but more accurate modeling has advantages and drawbacks which will be discussed in the experimental part. Note that the first three approaches can be applied with few information, as only the entrance count for each station is required, where the fourth approach requires user identification and the history of the users for a large time period to establish accurate individual models.

A last contribution of this work resides in the evaluation framework: the lack of supervision regarding network failures is critical to evaluate the proposed approach. To tackle this classical issue and to apprehend the advantages of each approach, three different protocols are proposed: the first one compares the ability of the approaches to detect *vanishing signals*, i.e. when no entrances are recorded for a small period of time; the second protocol explores qualitative results comparing the detected anomalies to a list of all particular days in the studied period (Christmas, Parisian terrorist attacks, Conference Of Parties -COP 21-, . . .). A quantitative study is presented using the messages emitted by the official Parisian transport authority (RATP) twitter account which reports many operating incidents. Those messages are preprocessed and used as a distant evaluation [17]. Finally, we propose a series of experiments on a toy dataset. This approach allows us to study our proposals strengths and weaknesses on a controlled environment.

The paper is organized as follows: Section 3 presents the context of the proposed work and usual anomaly detection approaches in Machine Learning; Section 4 introduces notations and the proposed models; Section 5 describes

[2] STIF Parisian authority estimates that about 30% of data are missing.

the protocol settings, the used datasets and the experiments conducted to assess the performances of the proposed models.

2. Related work

We first list interesting applications relying on smart card logs before focusing on anomaly detection. Finally, we propose a short review on nonnegative matrix factorization, demonstrating its interest for log analysis.

2.1. Smart card logs

Historically, ground survey was the only tool to monitor the use of a transportation system. But surveys have three main caveats: they are expensive, especially in the case of a wide transportation network covering a large urban space; due to the cost, they are rarely conducted and thus, are not able to reflect quick shifts of habits, temporary phenomena, or the changing dynamic of the network; at last, they are able to capture frequent recurrent habits efficiently but lack of precision for occasional trips due to the limits of the statistical tools. Since last decades, the growing use of smart cards in transportation networks offers a rich alternative to characterize transportation networks by exploiting the log data of the cards indicating when and where card owners use the network [18]. However, as other data coming from sensors (individual GPS traces, road sensors for traffic analysis, [4, 19]), smart card logs are massive, noisy and incomplete. Some category of users are not equipped with cards: tourist or occasional visitors may use tickets which aren't included in those data. Sensor failures and fraudsters are other common sources of noise. Moreover, log exploitation to construct accurate models is hard due to the randomness in the user's behavior and the lack of semantic information in the log : we know neither the reason of a trip, nor the destination[3]. Those difficulties have been first tackled by engineering robust statistical features designed with domain experts to demonstrate the relevance of log data to perform classical tasks as Origin/Destination matrices modeling the public urban transportation dynamic [20]. A common approach consists to aggregate data spatially, temporally and/or by groups of users to exhibit robust profiles: for example, [21] extracts various indicators (entrance time, number of bus stops use. . .) to model the use of Gatineau's (Quebec) transportation system. Such models with handcrafted features are competitive with ground survey techniques but requires fine expertise and are not able to exploit the whole information contained in log data. Hence, Machine Learning techniques have been introduced to model hidden signals and exhibit latent temporal and spatial profiles: for instance, [22] proposes a Gaussian mixture clustering approach to characterize users of the Rennes (France) urban transportation system according to their temporal behavior; [23] uses a Matrix Factorization model in the same purpose.

2.2. Anomaly detection

Anomaly detection has been widely studied in Machine Learning [8] in many applications: for instances in cyber security and intrusion detection [9], in fraud detection [10], in industry [24] or either in traffic applications as congestion detection [4]. The most common definition of an anomaly is a behavior which does not correspond to the considered model of the system [25]. Thus, detecting an anomaly involves generally two steps: inferring a model of the normal behavior of the system and considering a kind of distance or similarity between a given behavior and the model. The model can be a statistical parametric model when experts are able to design it and, in this case, statistical tests can be used to detect abnormal behaviors [26]. In some applications, labels are available indicating the normality or not of the behavior. In this case, usual supervised Machine Learning algorithms can be used to identify meaningful patterns: Neural Networks [27, 28] with various architectures (Recurrent Neural Network [11], Convolutional Neural Network [29]), Support Vector Machine [30, 31], decision trees [32]. However, due to their nature, anomalies are generally very uncommon in the datasets and supervised approaches are hindered by this unbalanced proportion. When no labels are available, unsupervised approaches can be used: density estimation based approach as parametric Gaussian mixtures have been proposed in [33], non-parametric Parzen Windows have been used in [34]. In both cases, behaviors belonging to low density regions are considered as abnormal. Other unsupervised approaches have been studied: [13] uses K-Means to infer normal behavior prototypes and a criterion based on the distance to classify the

[3] Parisian system is tap-in only, exits are not logged.

normality of the behavior; [35, 36] proposes to judge abnormal behavior with respect to the size of the clusters, small clusters denoting anomalies.

In case of sequential data and more precisely for time series, the principle of the approaches remains the same: first to infer a model of the normal behavior and then to compare a given behavior to the expected one. Modeling the time series can be done with naive algorithms as a moving average algorithm ([37]) or other regression algorithms (ARIMA [38], SVM [30], Neural Networks [39]). The distance between the prediction of the inferred model and the time series provides a measure of abnormality. The nature of the task and the invariance that the model has to capture will lead to the design of the model and the choice of the distance: a shape based distance can be used if 1) time series have unequal lengths (Dynamic Time Warping [40]) 2) times series are power comparable (subsequences matching [16, 41]). But those methods are computationally expensive and cannot tackle effectively high noise level.

Regarding smart cards logs, two main critical aspects have to be considered when designing a anomaly detection algorithm. Those data are noisy and often incomplete. On top of that, stations are heterogeneous regarding the signal power (some stations are big hub as others are small local stations). On the contrary, one aspect of the signals simplifies our problem: all time series are described on a 24h basis enabling us to deal with fixed-size vectors. As a consequence, we choose to focus on robust models exploiting vectorial inputs.

2.3. Nonnegative Matrix Factorization (NMF)

NMF is a common model for matrix decomposition. It decomposes a nonnegative data matrix into two new nonnegative matrices : a dictionary matrix, and a code matrix. The dictionary matrix contains shared basis or patterns which will be used by each example to approximate the signal. The code matrix contains the associated weight for each data example and each basis. Finally, an example in the data matrix will be approximated by a weighted combination (code matrix) of the shared basis (dictionary matrix).

NMF has been introduced by [42] as a decomposition algorithm for multivariate data. It provides compact (low rank approximation), robust (mean squared error learning criteria) and understandable (reconstruction is a sum of nonnegative patterns) representations. It has been particularly used in signal processing field [43, 44], but has been proved useful in other domains as recommender system [45] and image processing [46].

In the context of transportation systems, NMF has been used to analyze the network activity [47] or the user activity [48, 23, 5]: [47] proposes NMF approaches to decompose the aggregate taxi traffic flow at each location of Shanghai City (China). They model each location as a time series of traffic volume and decompose this matrix to find shared behaviors. Once this decomposition is learned, locations can be gathered according to their similarity. [48] represents each user of Rouen (France) urban transportation system as the mean of the number of validations for each day of the week and each hour of the day. A decomposition is learned using a NMF model. Then, they use the new representation of each user (code matrix) to cluster those users, allowing them to discover users that share similar behavior.

Several proposals already exist to impose a time consistency on the atoms of the NMF dictionary. [49] presents a smooth constrains to enforce smoothness along time frames in the context of audio source separation and uses a EM algorithm to learn this decomposition according to the constrain. Moreover, [50] presents a similar constrain, and uses a gradient descent learning process to learn a smooth dictionary in the context of EEG analysis. In this article, we use NMF decomposition to tackle noise in smart card logs and obtain robust, compact and meaningful representations. As a consequence, we deal with time series and have to face the time consistency issue. Our proposal -enforcing unimodal atoms in the dictionary- not only solve this problem but also lead us to learn atoms with a limited time support. Indeed, preventing any upturn in our atoms mechanically reduces their supports; thus, most atoms describe specific day periods corresponding to user activities, which is very interesting in our use case.

3. Models

The most common form of a smart card log is a triplet $(user, station, time)$ indicating the place and the time of the entrance of the user in the network[4]. We will consider in the following $L = \{\ldots, \ell_k = (user, station, time), \ldots\}$ the set of the collected logs of the network. As working hypothesis, we assume a weekly temporal stationarity of

[4] As most of transportation networks, the Parisian metro network is equipped with a tap-in smart card system; exits are not logged.

the network: each station has a specific behavior depending on the day of the week; as a consequence, we focus on (day, station) couples and represent them as a temporal aggregation of localized logs: the vector $\mathbf{x}_{s,i} \in \mathbb{N}^T$ gathers the activity for a station s during the i-th day. For all the experiments in this paper, we will consider a 91-day period (3 months, or 13 weeks) and choose a temporal discretization step of 1 minute (i.e., $T = 1440$ intervals for 24h). Each cell $x_{s,i}(t)$ corresponds to the number of entry-logs in the period. We denote by $X \in \mathbb{N}^{N \times T}$ the matrix gathering all $\mathbf{x}_{s,i}$. In this series of experiments, we focus on $\mathcal{S}$ stations. In the real world dataset, we focus on the Parisian metro network which has $\mathcal{S} = 300$; thus, we are considering $N = 91 \times 300 = 27300$ objects.

3.1. Baseline (BL) & Normalized Baseline (NBL)

Anomaly detection algorithms for time series are based generally on a distance to a regular regime [3]. We aim to detect outliers according to a ”normal” behavior. Two main strategy exist here : First, we can rely only on data and use kernel based methods (K-Means, K-NN, etc) to identify those ”normal” behaviors ([16]). Or, in the context of urban transportation system, a classical hypothesis to identify the regular regime is to consider a periodicity in the observations. This approach relies on the same methods as general ones but is more specialized, by adding some expert knowledge. Thus, we provide two very strong baselines to be compared with. We consider, in the following, a week periodicity, i.e. we assume that the station activity has a similar behavior every Sunday, Monday,...The objective in this context is to learn an aggregated reference model per couple (station,day of the week), i.e. 7 models per station. We compute

$$\bar{\mathbf{x}}_{s,d} \in \mathbb{N}^T = \frac{1}{13} \sum_{i|i \ mod \ 7=d} \mathbf{x}_{s,i} \qquad \text{for } d \in \{1, \ldots, 7\} \tag{1}$$

and obtain $N' = 7 \times 300 = 2100$ averaged references corresponding to the days of the week. We denote by $\bar{X} \in \mathbb{R}^{N' \times T}$ the matrix gathering all references. Then, we define an anomaly score function based on the L_1 distance between a couple and its associated reference:

$$\text{score}(s, i) = \|\mathbf{x}_{s,i} - \bar{\mathbf{x}}_{s,d_i}\|_1 = \sum_t |x_{s,i}(t) - \bar{x}_{s,d_i}(t)| \tag{2}$$

Those matrices provide a classical anomaly measure by considering the difference between the average behavior of a station and the real signal of the station for a given day i.

Such a modeling is sensitive to global amount of checking for a particular day; it is suitable to specific calendar day detection like bank holiday. In order to detect fine-grained anomalies, we propose a second baseline relying on a normalized version of the $\mathbf{x}$: $\mathbf{x}^\dagger_{s,i} = \mathbf{x}_{s,i}/\|\mathbf{x}_{s,i}\|_1$ and their associated normalized references $\bar{\mathbf{x}}^\dagger_{s,d}$. Each $\mathbf{x}^\dagger$ can be seen as the probability density function (pdf) giving the probability to observe a log at time t for a given couple (s, i). The previous L_1 anomaly score were also used for this representation to detect anomalies.

3.2. Nonnegative Matrix Factorization (NMF)

In this third approach, in order to increase the robustness with respect to the baselines, a supplementary assumption is made: the generic behavior of a station can be divided into few weighted common temporal patterns shared by all the network. Nonnegative Matrix Factorization (NMF) is one robust and flexible latent decomposition algorithm adapted to nonnegative datasets [51]. The presented approach uses such a decomposition algorithm on normalized data representation with an unimodality constraint.

Standard NMF. The idea consists in learning both a dictionary $D \in \mathbb{R}^{Z \times T}$ made of Z atoms $\mathbf{a}_z \in \mathbb{R}^T$ and the associated reconstruction code matrix $W \in \mathbb{R}^{N \times Z}$ so as to obtain

$$\mathbf{x}^\dagger_{s,i} \approx \sum_z w_{s,i}(z)\mathbf{a}_z, \qquad \mathbf{x}^\dagger_{s,i} \in \mathbb{R}^T \tag{3}$$

where $\mathbf{w}_{s,i} \in \mathbb{R}^Z$ is the weight vector associated to $\mathbf{x}^\dagger_{s,i}$. In addition to the general formulation of the regularized learning problem, we introduce a sparsity enforcement term. Finally the learning problem is the following one:

$$\text{argmin}_{W,D} \left\|X^\dagger - WD\right\|_F + \lambda_W \|W\|_F \tag{4}$$

It is optimized using the multiplicative update rules introduced by [51].

Two-step fast learning procedure. In our case, we divide the learning process into two steps to improve both speed and robustness:

1. $\bar{W}$ and D are learned on the reference matrix $\bar{X}^{\dagger}$ so as to obtain robust atoms quickly.
2. Once the dictionary D is fixed, W is learned by considering N independent reconstruction problems corresponding to the $\mathbf{x}^{\dagger}_{s,i}$ using the $\bar{\mathbf{x}}_{s,d_i}$ representation as initialization to enforce the use of same atoms for same days.

Such decomposition allows : 1) to learn a dictionary over the whole dataset, increasing the denoising power of the approach with respect to local aggregation; 2) to compress the representations of our objects by reducing the dimension of the description vectors from $T = 1440$ time steps to Z atoms; on top of the industrial interest of reducing the data size to store, please note that compression and denoising of raw signals are directly linked according to the minimum description length principle [52]; 3) to project all couples (station,day) in a unified non-orthogonal base to provide a better understanding of the differences.

Time-consistent NMF. The latter point, regarding the interpretation of the decomposition, motivates us to introduce modification in the original NMF formulation. Indeed, solving NMF (eq. 4), turn out to consider each feature of X as independent whereas in our case, the different features correspond to time steps which are clearly linked. Thus, NMF leads to the decomposition presented in Fig. 1 where several atoms describe both the morning and evening peaks (high energy regions). We also note that feature independence leads to a lack of smoothness in both the atoms and the reconstruction.

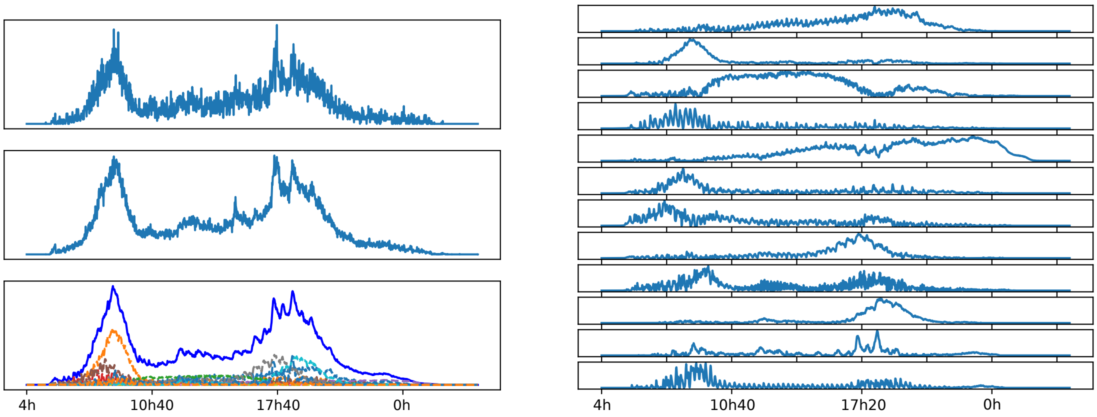


Figure 1: [left] (a) 09/10/2015 for station *Marcel Sembat*, (b) Averaged Wednesday model for *Marcel Sembat*, (c) NMF reconstruction of the first distribution. [right] classic NMF atoms examples extracted from the dictionary.

In order to enforce time consistency, we impose every atom of the dictionary to be unimodal (cf Fig. 2). This constraint is harder than the smoothness one that we have presented in the state-of-the-art section. It comes from the hypothesis that the logs we observe in the network comes from (unobserved) activities. Those activities are time located and our dictionary must be forced to model each activity as one atom. As a consequence of the unimodal constrain, we improve robustness by enforcing continuity between successive time steps. We also significantly simplify the interpretation of the decomposition, each atom tending to correspond to a single behavior (e.g. going to the office in the morning).

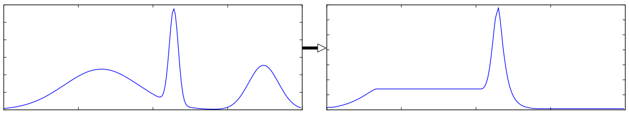


Figure 2: Unimodality constraint procedure illustration on a toy problem. [Left] Before constrain, [Right] After constrain. Atoms must be increasing until atom maximum, then it must be decreasing.

The procedure is very simple, due to the high flexibility of the original iterative NMF optimization process. In the original algorithm [51], W and D are alternatively updated -each sub-problem being convex-. We simply introduce a constraint step inside the loop where every atom is reshaped, starting from its highest value and smoothed so as keeping sign-constant derivative on both atom side (Fig. 2) :

$$\forall t \in [1, t^\star],\ d(t) = \min\left(d(t+1), d(t)\right) \tag{5}$$

$$\forall t \in [t^\star, T],\ d(t) = \min\left(d(t-1), d(t)\right) \tag{6}$$

where d is each dictionary atoms and $t^\star = arg\_max(d)$.

Computationally, this procedure is pretty light. For each atom in the dictionary, we pruned every value that break this constrain. This is achieved in $O(ZT)$. Obviously, the shape constraint unsettles the optimization procedure during the first iterations; however, the atoms quickly converge to their final shapes and the general gradient descent profile of the cost function (eq. 4) is roughly the same for the 2 versions of NMF. Fig. 3 illustrates this phenomenon. It also points out that our constraint increases the convergence speed after few iterations; then our model reaches a plateau and it is not able to reconstruct the noise: this fact explains why standard NMF has a better reconstruction error at the end of the iteration (we use a log scale to magnify the readability, but the two curves are finally very close in terms of reconstruction error).

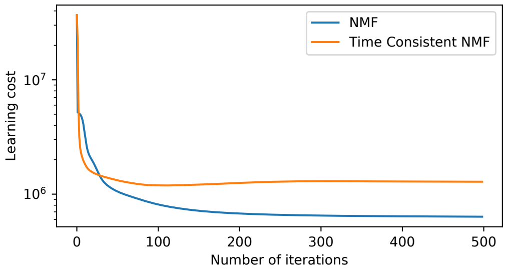


Figure 3: Learning cost evolution during training procedure.

The interest of the approach becomes clear on the example illustrated by Fig. 4: as intended, we obtain unimodal and meaningful atoms while preserving a comparable value for the optimal cost (see 3). Once stated that the reconstruction is smooth and robust, we will work directly in the compact latent space.

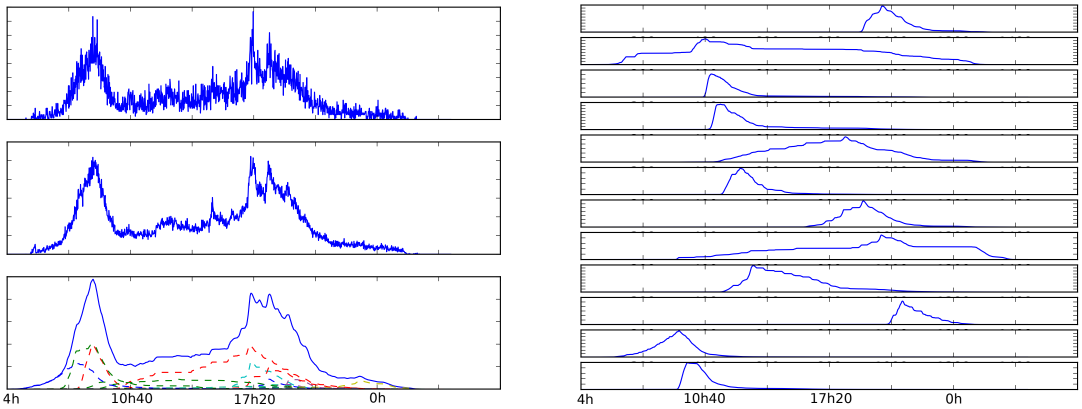


Figure 4: [left] (a) 09/10/2015 for station *Marcel Sembat*, (b) Averaged Wednesday model for *Marcel Sembat*, (c) NMF reconstruction of the first distribution. [right] Unimodal atom examples extracted from the dictionary.

NMF anomaly scoring. In the NMF procedure, original signals are normalized (namely, we work on $\mathbf{x}^\dagger_{s,i}$). Then, we enforce normalization of both the atoms ($\forall t,\ \sum_t a(t) = 1$) and the codes ($\forall s, i,\ \sum_z w_{s,i}(z) = 1$). As a consequence, we can see **w** as a discrete p.d.f. To stay coherent with other approaches, we choose as a distance measure the L_1 distance. Thus, the anomaly score becomes:

$$\text{score}(s, i) = ||w_{s,i} - \bar{w}_{s,d_i}||_1 = \sum_z |w_{s,i}(z) - \bar{w}_{s,d_i}(z)| \tag{7}$$

3.3. Continuous user-based model

This fourth approach takes the anomaly detection problem from another angle: we aim at characterizing the anomaly at log-level depending on the habits of the user who emitted this log. Some previous works ([5] have shown that modeling at a user level allow better characterization once this model is aggregated over the all system.

Individual models:. A first, we compute the likelihoods of every logs with respect to a spatial multinomial user model; then abnormal logs are identified and aggregated spatially and temporally; the analysis of the aggregated abnormal signal is used to detect abnormal (i.e. high) levels of anomalies. This fine-grained approach is much heavier than the previous ones and we have to overcome two main issues: users' models have to be learned on very few data and likelihood will be very sensitive to the log densities. As a matter of fact, the model risks to overfit user habits and to detect anomalies in every low-density period (such as midday and weekend).

To tackle users modeling while minimizing the overfitting risk, a very simple multinomial model is considered: it is purely spatial. In other words, we assume that an anomaly on the network will force the user to log in an unusual place. Thus, for a log $\ell = (u, s, t)$, its (spatial) probability is defined as:

$$p(s'|u') = \frac{|\{\ell \mid (u = u', s = s', t)\}|}{|\{\ell \mid (u = u', s, t)\}|} \tag{8}$$

In that way, the log probability distribution corresponds to the frequency of the stations in the user records. After several tests on joint spatio-temporal modeling, we chose to discard the time factor. Indeed, the high variability of the log amount along the day introduce a strong bias in the likelihood estimation: one log occurring at 15h00 (off-peak hours) will be automatically more abnormal than a 8h30 log (rush hour). Finally, we extract abnormal logs from all logs (i.e. all logs labeled with a low likelihood). In our experiments, we set a threshold and extract, for each user, only his 10% less likely logs.

Spatial & temporal aggregation. The transfer from ℓ likelihood to anomaly detection is not straightforward. An aggregation over the network has to be performed and next a post-processing step to smooth the computed signal by introducing a temporal diffusion around each log: the proposed model uses aggregation station by station and apply a Gaussian kernel as a time filter. The result is continuous modeling of the anomaly count (CMA). For a given station s', it can be written as:

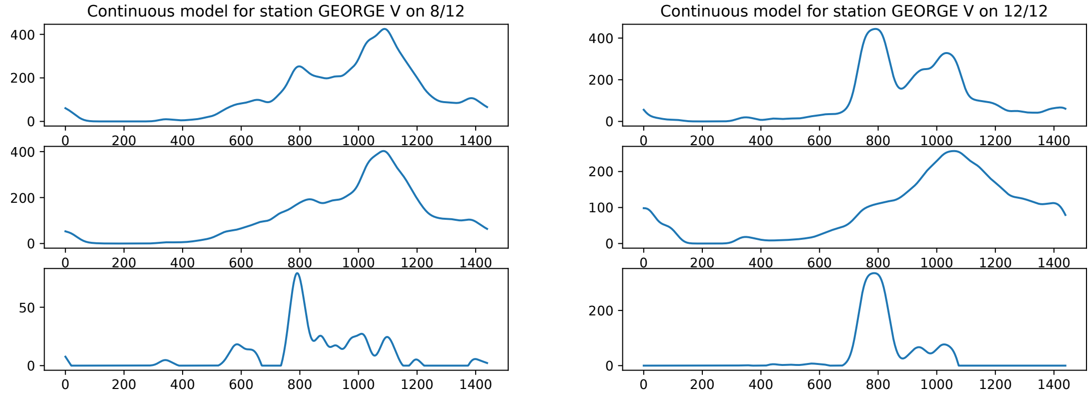
$$CMA_{s'}(t) = \sum_{\{\ell \mid (s=s',u',t')\}} p(s'|u') \circledast \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{||t - t'||^2}{2\sigma^2}\right) \tag{9}$$

However, as shown on Fig. 5, the model is systematically trapped by low density periods of the day when users are less predictable (midday, evening). In order to counter this effect, a differential approach is used as in the first approach focusing on situations *more abnormal as usual*. First, a reference CMA score is inferred for each of the 7 week days $\overline{CMA}_{s,d}(t)$; then, a deviation between $CMA_s(t)$ and $\overline{CMA}_{s,d}(t)$[5] is computed. In the resulting (and still continuous) model, the anomalies are detected by identifying most significant peaks[6].

This fine-grained method enables us to extract anomalies at any time granularity (minute, day, week, . . .). Hence, it offers more possibilities than previous approaches. Another important difference is in the detection mechanism based on a collective change of habit of users, but not sensible to absence of user's check-ins: in case of no trips for a user, no anomaly is detected. As consequence, the model should be robust to volumetric variation but inadequate to detect global anomalies as a fall of the global use of the network, holiday periods, etc.

[5] Obviously in $CMA_s(t)$, for each time t, we have to know the associate day d to perform a relevant comparison.

[6] Negative part of the signal is not considered: indeed, it would be difficult to analyze a *too normal* situation



(a) Tuesday

(b) Saturday

Figure 5: Continuous model for 2 days. [top] abnormality time series for station *Georges V*, [middle] Reference model for *Georges V*, [bottom] time series deviation from reference model for *Georges V*

3.4. Scalability of the propositions

Finally, we have to discuss the scalability of the proposed methods: Table 1 presents the computations costs for each model. Those costs correspond to the learning steps only, we will not discuss the anomaly scoring here (as this operation depends on the distance measure used). Obviously, the two baselines are very light to compute: they consist in a mean computation for each station and each week day. NMF starts from the normalized baseline and then learns a dictionary of size Z, each atom being defined on T time step: this iterative training procedure is far more expensive but remains reasonable (the complexity is linear with respect to all the terms). On top of that, NMF mainly relies on matrix operations that can be easily parallelized and/or distributed. The last model -Continuous User Based- demands a lot more calculation (usually in a transportation system, there is a huge number of logs) and memory (to store each user observation's probability for each station).

	BL	NBL	NMF	CM
Complexity	$O\big(L + dT\big)$	$O\big(L + 2dT\big)$	$O\big(L + 2dT + It \cdot d(2TZ) + ItTZ\big)$	$O\big(2L + 2US_u + WinL + 2dT\big)$

Table 1: Computation cost by models. With L the number of logs, d the number of days, T the number of timestamps in a day, U the number of users, S_u the mean number of stations per users, Win the size of the convolution window for CM and It the number of learning iteration for the NMF

Regarding the inference step, our conclusions are very close; indeed, for the baselines, this step is straightforward, we only need to compute the distance between the day-station vector and the reference. NMF requires a mapping of the day-station vector on the existing dictionary which remains manageable. On the contrary, CM inference is almost as expensive as the training.

4. Experiments

In this section, we first describe and exploit a toy dataset to obtain a better understanding of the behaviors of our approaches. Then we present the real world smart card dataset, the models parameters as well as the twitter dataset that we collected for the evaluation procedure; thus, we compare the performances of the different models over three tasks: vanishing signal detection, operating incidents reported from the official Parisian authority twitter account, and special days detection.

4.1. Toy dataset

In order to study each method strengths and limits in a controlled environment, we design a generative model able to create logs from a set of users over a set of stations. Inspired from a real urban transportation system, we make the following assumptions:

- each log is the realization of an activity,
- activities are shared by a group of users and are time recurring (with a varying time period),
- each activity for each user is perform on a specific station,
- stations are multi-scale (on different activities): some are hubs, others are small stations; some mainly serve residential sector while others are located in an office district.

This dataset allows us to add anomalies and control the experiments parameters like noise level and stations multi-scaling.

Hence, our generative model is activity based. We define an activity as a probability density function (p.d.f.) over time; all p.d.f. follow the normal distribution. Each activity can be realized by each user at various times (user i leaves to work at 8h a.m. while user j leaves to work at 9h a.m.). Each activity is also associated to a periodicity (day, week, month) and to a spatial multinomial p.d.f. (stations have different sizes and different roles. Formally, we define an activity a as :

$$a = \{\mu_a, \sigma_a, fq_a, sts_a\} \tag{10}$$

where fq_a is the frequency, μ_a the time realization mean, σ_a the standard deviation, and sts_a the set of stations where this activity can occur. We generate for 100 000 users and 100 stations with 9 activities : 5 daily, 3 weekly and 1 monthly.

We first pick a user, then a set of activities; for each activity, we draw an individual $\mu_u \sim \mathcal{N}(\mu_a, \sigma_a)$. Log times are generated randomly according to $\mathcal{N}(\mu_u, \sigma_u)$, where σ_u is a constant. The log is then located according to sts_a. Then the operation is repeated with respect to the time period fq_a. Finally, we define the forgetting probability as 0.1 (ie, one time over ten, the log is not stored) to model both sensor and database malfunctions. On top of that a uniform noise is added: some logs are drawn from the uniform distribution (both on temporal and spatial aspects) and added to the dataset. At the end of the procedure, our synthetic dataset is made of about 14 millions logs (depending on the noise level).

Obviously, the NMF will benefit from this generative procedure: the above mentioned assumptions are close to those made in our NMF-framework. However, we also hope that those assumptions jibe with reality and correspond to the true activities. We introduce three kinds of anomalies in the data: the local shift –when, for a particular moment in a particular station, logs are temporally shifted–, the activity volume change –when for a particular moment in a particular station, a part of the traffic is missing– and the station closing. For the latter, we consider that users still log-in in a station but then, they realize that this station is *disabled* and, with a given probability, they log-in in an other near-by station.

4.1.1. Anomaly detection

All couples day-station are modeled according to the methods described in section 4. Then, we rank all couples with respect to their references and consider the N furthest as anomalies. With the toy dataset, we choose N corresponding to a given percentage K of the data and we compare all the methods using the precision@K measure. Here, we choose $K = 10\%$ and so $N = 920$. All results are given as the mean value over 5 runs.

4.1.2. Local shift

We assume that a local incident causes a delay on a particular station. We introduce in our generated dataset 157 anomalies corresponding to those local shifts. Each anomaly is defined by a station, a starting and ending moment, a shift delay and a shift probability. We delay (by the shift delay parameter) a proportion (shift probability) of all logs occurring in the anomaly moment on the particular station. We measure our ability to detect with respect to the length of the incident, the proportion of delayed logs and the probability of forgetting logs (figure **??**).

This task, inspired by real situations, suits perfectly to the NMF framework: even a small shift should generate a wide change in the code matrix representing the station. In practice, NMF is indeed the only model to catch efficiently this phenomenon. Even in case of a strong shift, other approaches remain blind due to the weak amplitude of the movement. We obtain stable results even for a high level of log forgetting probability.

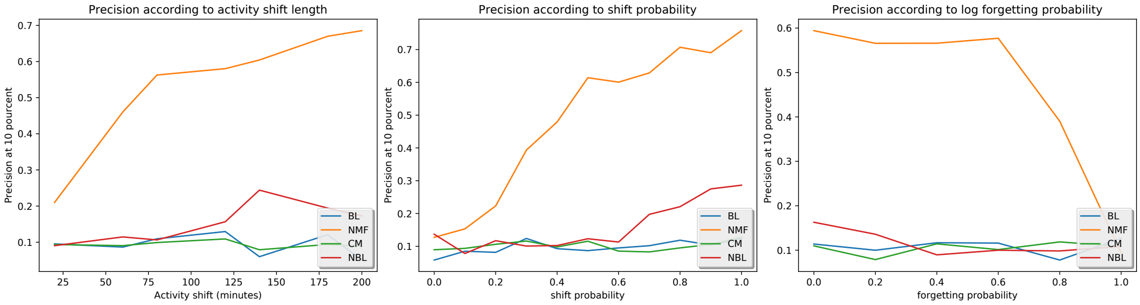


Figure 6: [Left] Precision at 10% for activity shift detection w.r.t. shift length with log shift probability level of 70%. [Middle] For a mean shift of 120 minutes (standard variation of 20 minutes), precision at 10% for activity shift detection w.r.t. the proportion of log actually delayed. [Right] For a mean shift of 120 minutes (standard variation of 20 minutes), precision at 10% for activity shift detection w.r.t. the proportion of log not stored (system malfunctions)

4.1.3. *Activity volume change*

Our second anomaly corresponds to a special day where the users' behaviors are modified. We assume that for a particular moment in a particular station, a proportion of logs does not occur. In our experiments, we introduce 313 anomalies. Obviously, we expect NMF to overcome significantly other methods since the volumetric change should appear straightaway in the latent representation of the affected stations.

Indeed, Figure **??** shows that the phenomenon remains undetectable up to 20% of change; then, BL and CM are still unresponsive –curves correspond to random detectors– while NMF and NBL detect some anomalies. As it is expected, NMF obtains good performances on this task. Regarding CM, no geographic changes impacts the user: as a consequence, the model is not able to perceive the modification. The BL model is disadvantaged by the fact that the activity reweighting may affect only a small proportion of the log with respect to the day amount.

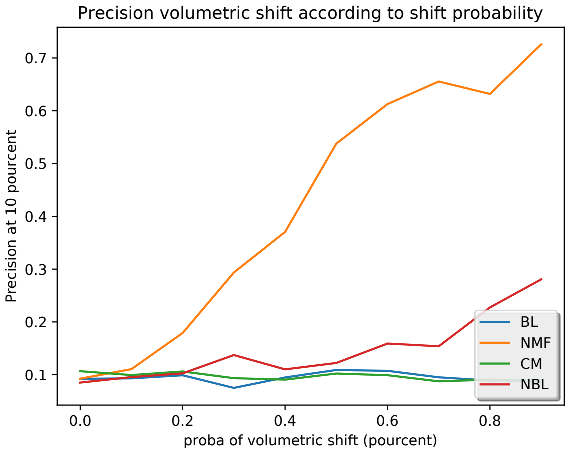


Figure 7: Precision at 10% for activity shift according to the proportion of volumetric shift

4.1.4. *Station closing*

The last simulated incident corresponds to a station closing. We assume that users first validate in the disabled station –a classical scenario in the Parisian network– and then emerge and choose another close station with a given probability. When no traffic is deported, the incident is undetectable; when the deportation rate increases CM is clearly the best model to catch the phenomenon. In our experiments, we introduce 40 station closing anomalies. Each of those, deporting the traffic on 3 other "near-by" stations. To evaluate model's precisions, we extract the number of anomalies with at least one detection on the "near-by" stations. Those results are consistent since the individual user modeling is sensitive to the station where the log occurs: even a slight proportion of unusual log location raises an alarm. Surprisingly, NMF performs well on this task: this is due to both the denoising ability of the method and the limited time support of the atoms. Indeed, the log deportation generates a time located little bump that is caught by one atom: the difference with respect to the reference rapidly becomes significant and an alarm is raised. On the contrary, the traffic growth around the impacted station is too small –considering the whole day– for a detection with BL or NBL.

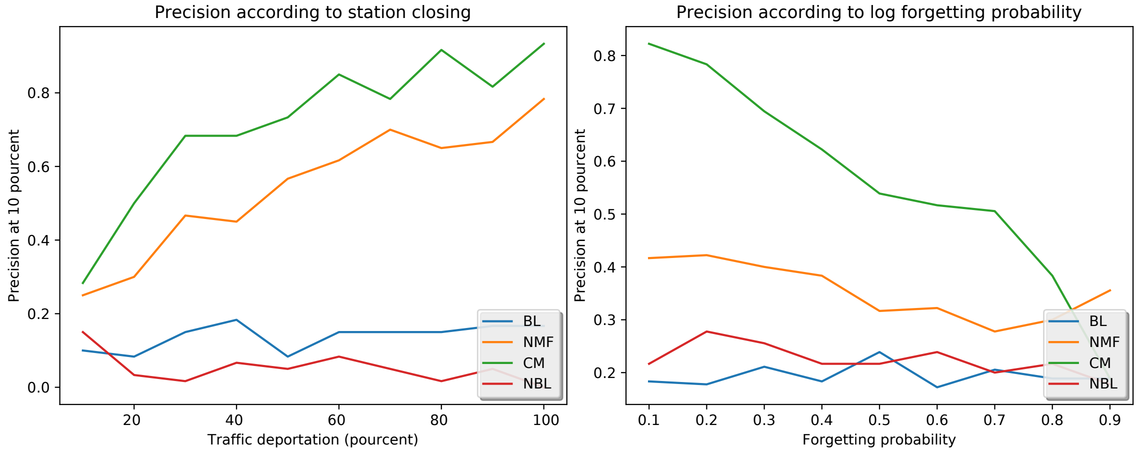


Figure 8: Precision at 10% for traffic deportation incident. Left figure gives the evolution of the detection precision w.r.t. the percentage of log observed in close stations. Right figure shows the performance course when the level of noise increases.

4.2. *Real world dataset, parameters and supervisions*

The smart card dataset counts more than 256 million logs and correspond to more than 3.3 million users on the Parisian metro network (300 stations)[7]. Those users only reflect a part of all possible users categories. Indeed, all users which uses tickets to access the transportation system are not logged here. So this dataset focuses mostly on parisian area population moves, ignoring most tourists and occasional visitors. It has been collected during the last 3 months of 2015 (October, November, December), which correspond to 91 days or whether 13 weeks.

Regarding models parameters, we choose for the NMF approach to extract 24 atoms. In the best of our knowledge, there exist no automatic algorithm to set the number of the NMF's atoms. This number had to be selected empirically. It seems large enough to cover a wide scope of possible behaviors, and small enough to limit the dictionary redundancy. For the continuous user-based model, we choose the day as time granularity for the anomalies extraction procedure. This allows us to easily compare our proposals, as all anomalies will be expressed in a day granularity.

As in classical studies [3], a first focus concerns special days. Considered period includes 8 events impacting the whole network[8]: Paris hosted the 21^{st} edition of the United Nations Climate Change Conference (COP21). This conference starts the November 30, 2015, and all public transportation system was free for this day. This implies a strong perturbation in the subway use. Then Christmas and New Year occur at the end of December. During those 3 days (December 24, 25 and 31), people usually spend time with their family and an unusual use of the transportation system is expected. The 11 November is a bank holiday in France, and transportation network use will also be perturbed. Finally, the fateful Bataclan terrorist attacks occur on November the 13^{th}, implying a huge flow of panicked people and during the next days, a strong drop in the logs.

We also identified sensors failures corresponding to a *vanishing signal*, when no logs are recorded for a station whereas it still works normally. Those abnormal situations, easy to detect, correspond to an abrupt drop of the energy in the log flow: our methods behave differently on this case. We distinguish 2 subsets: a small one gathering outage exceeding 1 hour (1214 events) and a larger one for interruptions larger than 30 minutes (1558 events).

[7] The metro network serves Paris and the inner band of its surrounding territories

[8] corresponding to $8 * 300 = 2400$ anomalies

Finally, the third evaluation protocol focus on *operating incidents*. The impact of those anomalies depends on the severity of the incident, ranging from interruptions of service with a local drastic fall of the station use to slowdowns of the traffic with less impact on the log data of the station but more diffuse in time and space. In order to compute a quantitative performance on this task, the Twitter account of the RATP (Parisian metro authority) was crawled and processed during the studied period. As a result, 362 operating incidents were identified together with their spatial and temporal information: the time-stamp of the alert message, the duration of the incident, the metro line concerned and the list of impacted stations. Considering the duration of the incident as an indicator of the incident severity allows to rank anomalies and to obtain a sorted list of ground truth incidents that we use in the following.

4.3. Reconstruction error

Model	L1 score	L2 score	L1 score Week	L1 score Week End	L2 score Week	L2 score Week End
NBL	0.46 (0.29)	0.0043 (0.05)	0.41 (0.27)	0.58 (0.31)	0.0016 (0.017)	0.011 (0.08)
NMF	0.45 (0.25)	0.0044 (0.05)	0.41 (0.23)	0.56 (0.27)	0.0015 (0.016)	0.011 (0.08)

Table 2: Quantitative results on data reconstructions. Mean (standard deviation)

As our three first approaches (baselines and NMF) can be seen as generative models, we propose to measure and compare the reconstruction errors associated to our models. In practice, we focus on NMF and normalized baseline: indeed, raw data logs are not directly comparable to normalized figures. Table 2 provides a summary of reconstruction cost for those two approaches. We clearly see that their reconstruction abilities are very close. We note that NMF is slightly more robust than NBL, as it as a lower standard deviation for all reconstruction problem. Moreover, we bring to light a well known phenomenon: the week activity in our transportation network is more regular than the week-end activity; this can be easily explained by the fact that most users works -regularly- on the week and go out for leisure -irregularly- during the week-end.

The reconstruction abilities -very similar- should be considered with respect to number of parameters required to estimate the baselines. In that sense, the NMF compression skills is remarkable. This could lead to future works in other fields.

4.4. Vanishing signal detection

Vanishing events can be labeled easily by identifying time windows with a zero signal in the raw representation of each couple (station, day). To study the ability of our model and baselines to detect those vanishing signals, we propose an evaluation close to the bipartite ranking framework [53].

The following ROC curve is computed: each point corresponds to a threshold α on the anomaly score of the different models. Namely, the α% top ranked couples according to our model are considered as positive - presenting an anomaly - and the rest as negative; For each α, we plot the true positive rate (percentage of positive couples labeled as positive) w.r.t. the false positive rate (percentage of negative couples labeled as positive). The area under the curve gives the overall performance of the model. Fig. 12 shows the ROC curves for a minimal vanishing intervals respectively greater than 30 min (left) and 1 hour (right).

The standard baseline (BL) alarms are concentrated on atypical days and the model is not able to catch vanishing signals efficiently. Then, NMF perform as well as the normalize baseline (NBL) in both experiments. More than 80% of anomalies due to a vanishing signal of more than 30 minutes are detected at very first ranks for the NMF model compared to the 40% for the BL. Generally speaking, the denoising ability of the NMF and the sparse representation helps the NMF model to distinguish between irregularities due to the noise and real signal anomalies. Finally, we see that the continuous model (CM) focuses on other types of anomalies. This latter model is sensitive to abnormal logs but not to absence of logs. This is typically the kind of situation that generates *over-normality* alarms; dealing efficiently with those special cases remains should be studied in future work.

4.5. Distant evaluation with Twitter

This series of experiments uses the operating incident corpus collected from twitter as ground truth according to the metrics presented previously. The first experiment focuses on the detection of anomalies at a day scale. The second experiment conducts two specific analyses to determine the accuracy of the twitter events recognition with respect to the time period and the metro station where the incident occurs.

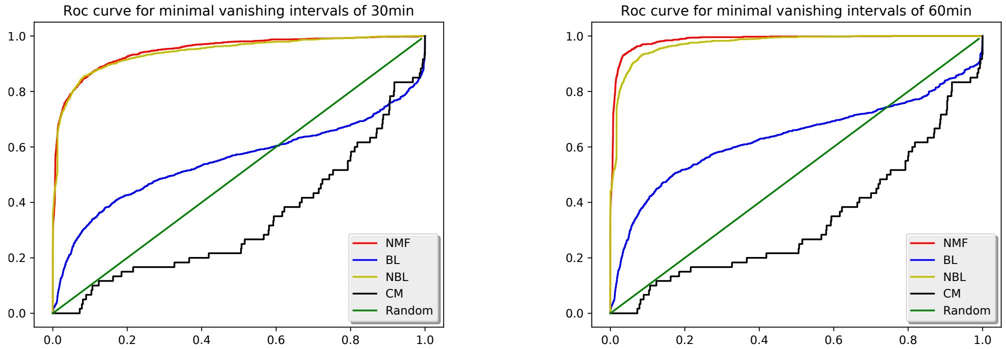


Figure 9: ROC curves for a minimal vanishing intervals of 30min (left) and 1 hour (right).

Abnormal day detection. Results of each approach are aggregated by days over the whole network and abnormal days are ranked according to this aggregation. The obtained ranking is compared to the aggregated twitter abnormality day ranking: for a given day, all duration of events occurring this day are summed and days are ranked accordingly to this value. Fig. 13 (left) presents the ROC curves showing the capacity of each models to retrieve the 20% more abnormal days.

Those results should be read keeping in mind that our models work on different representations of the logs. Standard baseline (BL) catch strong volume gaps: it obtains good performances by retrieving obvious events. Normalized baseline (NBL) and NMF operate on the same data: once all volumes normalized, they are searching for slight behavioral changes. This general similarity in the raised alarms is pointed out on Fig. 13 (right). On that aspect of the detection, NMF clearly outperformed NBL (Fig. 13, left).

The continuous model (CM) relies on a completely different paradigm. As a consequence, it detects other incidents than NMF (Fig. 13, right). In particular, it does not detect signal vanishing while this kind of anomaly is caught very early by the other approaches. The same phenomenon occurs for special days: when a user does not go to work early in the morning, it causes an anomaly for BL, NBL and NMF... But the absence of signal does not impact CM. In terms of pure abnormal day detection according to the twitter ground truth, it outcomes all models: it is extremely efficient in catching slight modifications in the network records.

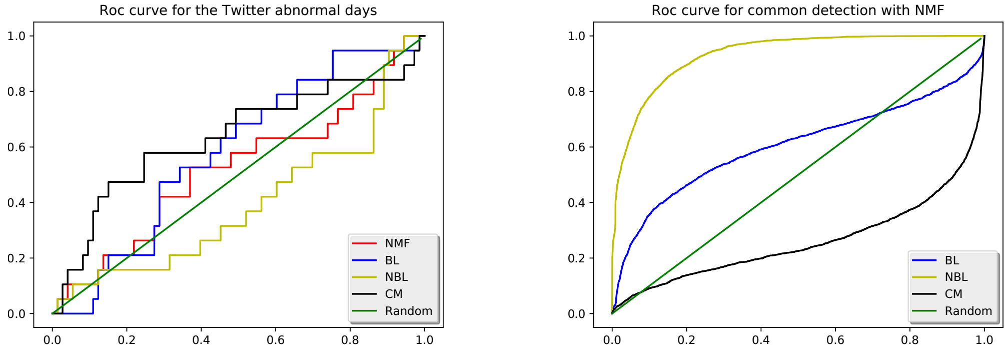


Figure 10: (left) ROC curves of abnormal day detections, blue: BL; yellow: NBL; red: NMF. (right) % of common detection w.r.t. the threshold of the detectors, red: BL *vs* NMF, blue: NBL *vs* NMF.

Geographical and temporal detections. Whereas the previous metrics was recall-oriented, this paragraph focuses on precision, if anomalies detected by the proposed approaches are able to explain the twitter events. The first 10% of couples (day,station) corresponding to the higher deviations according to each model is kept, and then we analyze different kind of matching between retrieved abnormal couples and twitter events: time matching, geographical matching and exact matching. In detail, if the detected anomaly occurs during the tweet incident, then the tweet is temporally explained, if the anomaly is raised on the problematic stations (i.e. mentioned by the tweet), it is geographically explained, finally if the anomaly is on the problematic station and is temporally explained, then the tweet is exactly explained.

Table 3 illustrates the results. First of all, good BL performance is balanced by the fact that it corresponds to major incident detection that are easier to catch. Then, NMF performs well on geographical aspects w.r.t NBL, but it is overcome on the temporal benchmark. Indeed, in order to determine the time location of the incident, we select the peak of the atom with the highest variation w.r.t. to the reference: given the support of the atom, we have too large an approximation. Finally, we see that CM outperforms other models for all tasks. CM is intrinsically a fine-grained model well suited for this series of experiments.

	BL	NBL	NMF	CM
Geographical accuracy	87.6%	42.5%	48.1%	**84.2**%
Temporal accuracy	69.9%	72.1%	42.2%	**92.0**%
Exact matching	0%	0%	0%	**16.3%**

Table 3: Accuracies of the 4 models to determine the metro stations impacted by the incidents (Geographical), the associated time period (Temporal) and both aspects (Exact). Incidents are extracted from the Twitter dataset.

4.6. Qualitative analysis

Finally, we investigate the semantic bias of each model. Starting from the 10% most powerful alarms of each model (2751 alarms), Table 4 gives the general precision of the models and the distribution of the alarms on the 3 categories detailed in section 5.2.

	BL	NBL	NMF	CM
# of anomalies	2751	2751	2751	2751
Special days	857	1378	**1774**	92
Vanishing signal	56	**315**	307	17
Twitter	780	88	126	**1037**
Precision	61.5%	64.7%	**80.2**%	41.7%

Table 4: Qualitative analysis of anomalies by models and types.

NMF offers a good compromise between all types of anomalies but at the cost of high level of imprecision, unable to detect the Twitter anomalies. This counter-performance is due essentially to a lack of temporal accuracy: as soon as the event is located in a limited time window (e.g. 15 minutes of traffic interruption), NMF (and NBL) don't effectively detect them. On the other hand, CM can't detect the special days and vanishing signal events, because if there's a few or no logs, then, there is no abnormal log, and so, no detection. On the contrary, it is well suited to detect "small" events like Twitter ones. It should be noted that the twitter events detected by BL and CM are generally not the same ones. BL still focuses on high energy phenomena whereas CM is robust to smaller events.

Regarding the general precision, NMF demonstrates it robustness by achieving more than 80% of relevant detections, far ahead of the other approaches. Due to its denoising and compressing ability, most of its deviation indeed corresponds to real events.

Obviously, we cannot be (and are not) fully comprehensive on anomalies that occur in the studied period. We believe that we cover the most of them, but it is not because an anomaly is not referenced by our ground truth sets, that it is not a "true" anomaly. We believe that this experiment provides a good insight of what types of anomaly are detected by each model.

Finally, we propose an overview of the models behaviors in Fig. 14. This illustrates the spatiotemporal distribution of the alarms for every model. We see that BL is focused on high energy stations (vertical lines) and specific days. Almost no detection arise outside this limited scope. On the contrary, CM detects local events with a clear deficiency regarding abnormal days. NBL and NMF represent both interesting compromise but NMF is clearly more robust to the general energy of the station (less vertical lines).

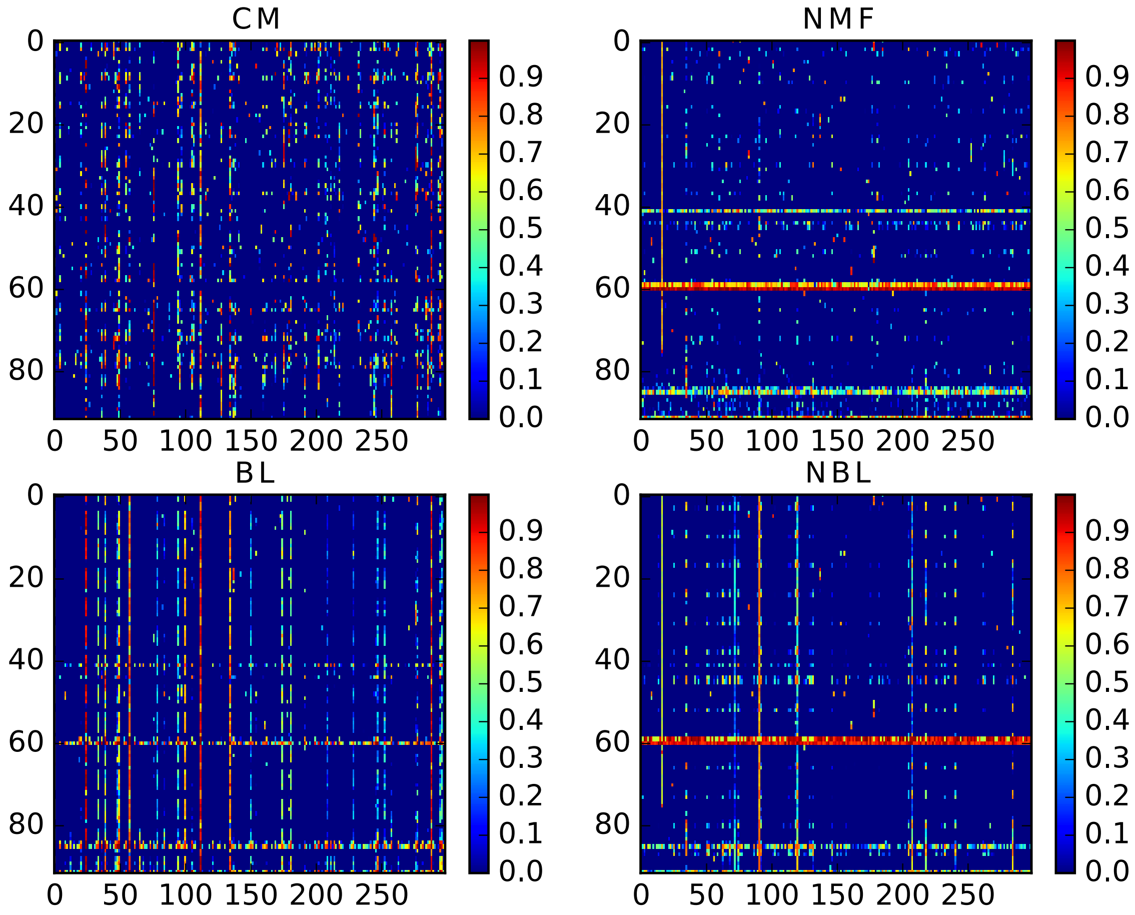


Figure 11: Spatio-temporal grids give the distribution of the 3000 most powerful alarms according to each model. For all plots, x-axis corresponds to the 300 stations and y-axis to the 91 days.

5. Conclusion

We tackled the anomaly detection problem in smart card log flows. We proposed a NMF approach, modified to enforce a time consistency constraint, and a user-based modeling. Both approaches show very interesting results with respect to standard strategies based on differential computation in the raw log domain. We evaluated our framework on a toy model as well as a real-world dataset. We also introduce an original distant evaluation scheme that enables us to show quantitative results on an unsupervised task.

The modified NMF approach is robust and provides very compact representations for station profiles. We demonstrate its superior ability to catch anomalies in noisy signals with respect to the baselines. Contrary to raw signal detectors and user based modeling, NMF gives very few false alarms: more than 80% of the alarms correspond to real events in our dataset.

The user-based approach is impressive at detecting small incidents reported in the twitter dataset. We demonstrate that it is very efficient to detect fine-grain changes in traffic even if it fails to detect obvious events like signal vanishing.

A first perspective of this work concerns the fusion of the different models. Indeed, we show clearly that each model (standard baseline, NMF, user-based) is specialized in a particular type of incident, respectively high energy signal distortions, special days implying behavior modifications and weak signal deviations. Fusing those approaches is a real perspective to build a strong versatile anomaly detector. Then a second perspective regard anomaly diffusion and characterization. As a matter of fact, the user-based model shows that we are able to reliably detect slight signal modification; this offers the perspective to analyze the evolution a single anomaly within the network. We could be able to characterize the dynamics of a perturbation and its radius.

Acknowledgment: The authors gratefully acknowledge the STIF regarding the exploitation of the dataset. This work was partially founded by the FUI AWACS grant.

[1] P. Newman. Why do we need a good public transport system. *Research Paper, Curtin University Sustainability Policy (CUSP) Institute.*, 2012.

[2] John A Black, Antonio Paez, and Putu A Suthanaya. Sustainable urban transportation: performance indicators and some analytical approaches. *Journal of urban planning and development*, 128(4):184–209, 2002.

[3] M. Trépanier, C. Morency, B. Agard, E. Descoimps, and J.S. Marcotte. Using smart card data to assess the impacts of weather on public transport user behavior. In *Conference on Advanced Systems for Public Transport*, 2012.

[4] Irina Ceapa, Chris Smith, and Licia Capra. Avoiding the crowds: understanding tube station congestion patterns from trip data. pages 134–141. ACM Knowledge Discovery and Data Mining (KDD) workshop on urban computing, 2012.

[5] M. Poussevin, E. Tonnelier, N. Baskiotis, V. Guigue, and P. Gallinari. Mining ticketing logs for usage characterization with nonnegative matrix factorization. *Lecture Notes in Computer Science (LNCS) Big Data Analytics in the Social and Ubiquitous Context*, 2016.

[6] S. Foell, G. Kortuem, R. Rawassizadeh, S. Phithakkitnukoon, M. Veloso, and C. Bento. Mining temporal patterns of transport behaviour for predicting future transport usage. In *Conference on Pervasive and ubiquitous computing adjunct publication*, 2013.

[7] T. Camacho, M. Foth, and A. Rakotonirainy. Pervasive technology and public transport: Opportunities beyond telematics. *IEEE Pervasive Computing*, 12(1), 2012.

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[9] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, Secondquarter 2016.

[10] Jarrod West and Maumita Bhattacharya. Intelligent financial fraud detection: A comprehensive review. *Computers and Security*, 57, 2016.

[11] Min Cheng, Qian Xu, Jianming Lv, Wenyin Liu, Qing Li, and Jianping Wang. Ms-lstm: A multi-scale lstm model for bgp anomaly detection. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–6, Nov 2016.

[12] Fabio Guigou, Pierre Collet, and Pierre Parrend. Anomaly detection and motif discovery in symbolic representations of time series. *CoRR*, abs/1704.05325, 2017.

[13] Ravi Ranjan and G. Sahoo. A new clustering approach for anomaly intrusion detection. *International Journal of Data Mining and Knowledge Management Process (IJDKP)*, 4, 2014.

[14] D. Liu, C. H. Lung, N. Seddigh, and B. Nandy. Network traffic anomaly detection using adaptive density-based fuzzy clustering. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 823–830, Sept 2014.

[15] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 430–433, Washington, DC, USA, 2004. IEEE Computer Society.

[16] V. Chandola and V. Kumar V. Mithal. A comparative evaluation of anomaly detection techniques for sequence data. In *In Proceedings of the 2008 8th IEEE International Conference on Data Mining (ICDM)*, pages 743–748, 2008.

[17] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[18] Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557 – 568, 2011.

[19] Ana I Gall and Fred L Hall. Distinguishing between incident congestion and recurrent congestion: a proposed logic. *Transportation Research Record*, (1232), 1989.

[20] Sui Tao, David Rohde, and Jonathan Corcoran. Examining the spatial–temporal dynamics of bus passenger travel behaviour using smart card data and the flow-comap. *Journal of Transport Geography*, 41:21–36, 2014.

[21] Catherine Morency, Martin Trépanier, and Bruno Agard. Measuring transit use variability with smart-card data. *Transport Policy*, 14(3):193 – 203, 2007.

[22] A. S. Briand, E. Côme, M. K. E. Mahrsi, and L. Oukhellou. A mixture model clustering approach for temporal passenger pattern characterization in public transport. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, Oct 2015.

[23] Emeric Tonnelier, Nicolas Baskiotis, Vincent Guigue, and Patrick Gallinari. Smart card in public transportation: designing a analysis system at the human scale. In *Intelligent Transportation Systems Conference (ITSC)*, Rio de Janeiro, Brazil, 2016. IEEE.

[24] Luis Martí, Nayat Sanchez-Pi, José Manuel Molina, and Ana Cristina Bicharra Garcia. Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 15(2):2774–2797, 2015.

[25] F. J. Anscombe and Irwin Guttman. Rejection of outliers. *Technometrics*, 2(2):123–147, May 1960.

[26] H. E. Solberg and A. Lahti. Detection of outliers in reference distributions: Performance of horn's algorithm. *Clinical Chem.*, 51(12):2326–2332, 2005.

[27] S. J. Hickinbotham and J. Austin. Novelty detection in airframe strain data. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 536–539, 2000.

[28] D. Dasgupta and N. S. Majumdar. Anomaly detection in multidimensional data using negative selection algorithm. In *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, CEC '02, pages 1039–1044, Washington, DC, USA, 2002. IEEE Computer Society.

[29] Shifu Zhou, Wei Shen, Dan Zeng, Mei Fang, Yuanwang Wei, and Zhijiang Zhang. Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes. *Signal Processing: Image Communication*, 47:358 – 368, 2016.

[30] Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 613–618, New York, NY, USA, 2003. ACM.

[31] M. Davy and S. Godsill. Detection of abrupt spectral changes using support vector machines, an application to audio signal segmentation. In *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[32] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct 2004.

[33] R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In *2009 12th International Conference on Information Fusion*, pages 756–763, July 2009.

[34] C. Chow and D.-Y Yeung. Parzen-window network intrusion detectors. In *In Proceedings of the 16th International Conference on Pattern Recognition*, volume 4. IEEE Computer Society, Washington, DC, USA, 40385, 2002.

[35] A. Pires and C. Santos-Pereira. Using clustering and robust estimators to detect outliers in multivariate data. In *In Proceedings of the International Conference on Robust Statistics*, 2005.

[36] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recogn. Lett.*, 24(9-10):1641–1650, June 2003.

[37] Sabyasachi Basu and Martin Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems*, 11(2):137–154, 2007.

[38] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An anomaly detection method for spacecraft using relevance vector learning. In *Advances in Knowledge Discovery and Data Mining*, volume 3518, page 785–790, 2005.

[39] David J. Hill and Barbara S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling and Software*, 25(9):1014 – 1022, 2010.

[40] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international con- ference on Knowledge discovery and data mining*, page 216–225, 2003.

[41] K. Sequeira and M. Zaki. Admit: Anomaly-based data mining for intrusions. In *Proceedings of the 8th ACM international conference on Knowledge dicovery and data mining (KDD)*, pages 386–395, 2002.

[42] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *In Neural Information Processing Systems (NIPS)*, pages 556–562. MIT Press, 2000.

[43] S. Ewert, B. Pardo, M. Mueller, and M. D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, May 2014.

[44] A. Ozerov and C. Fevotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, March 2010.

[45] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *International conference on Data mining (SDM)*, pages 549–553. SIAM, 2006.

[46] T. X. Luong, B. K. Kim, and S. Y. Lee. Color image processing based on nonnegative matrix factorization with convolutional neural network. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 2130–2135, July 2014.

[47] Chengbin Peng, Xiaogang Jin, Ka-Chun Wong, Meixia Shi, and Pietro Liò. Correction: Collective human mobility pattern from taxi trips in urban area. *PLOS ONE*, 7(8), 08 2012.

[48] Carel L. and Alquier P. Non-negative matrix factorization as a pre-processing tool for travelers temporal profiles clustering. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgique*, 2017.

[49] C. Févotte. Majorization-minimization algorithm for smooth itakura-saito nonnegative matrix factorization. *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011.

[50] Zhe Chen and Andrzej Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Laboratory for Advanced Brain Signal Processing*, 2004.

[51] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 2004.

[52] Peter D Grünwald. *The minimum description length principle*. MIT press - The Minimum Description Length Principle (Adaptive Computation and Machine Learning), 2007.

[53] S. Clémençon and N. Vayatis. Adaptive estimation of the optimal roc curve and a bipartite ranking algorithm. In *Algorithmic Learning Theory*. Springer, 2009.