Prédiction de pannes DSL par mesure passive sur des passerelles domestiques

Pascal Le Guyadec, Christoph Neumann

Prédiction de pannes DSL par mesure passive sur des passerelles domestiques

Pascal Le Guyadec, Christoph Neumann

Technicolor, Rennes, France

Les liens DSL peuvent subir des pannes sporadiques entraînant des déconnexions ou un accès Internet dégradé. Ces pannes sont à l'origine d'une expérience utilisateur négative et génèrent des coûts pour les fournisseurs d'accès Internet (FAI) via des appels d'assistance technique. La prédiction de pannes permet aux FAI de mettre en oeuvre des mesures proactives de mitigations de pannes. Dans cet article, nous discutons comment effectuer la prédiction en ligne de pannes DSL en utilisant des mesures passives faites de manière continue par des passerelles Internet domestiques. Contrairement aux travaux antérieurs, notre approche permet une prédiction fine des pannes, plusieurs heures ou jours avant leur apparition. En utilisant une collection longitudinale de métrologies DSL sur 2 ans dans 98 maisons nous montrons que nous prédisons les pannes avec de bonnes performances.

Mots-clefs : failure prediction, machine learning, passive measurements, DSL

1 Introduction

Digital Subscriber Line (DSL) is a broadband technology that is used in many countries to connect the homes of end-users to the Internet. DSL links are not always fully stable and can suffer from sporadic breakdowns, leading to disconnections or degraded access to the Internet. Such failures are the cause of negative user experience and generate costs to operators through helpdesk calls. Jin et al. [J^{+}10] highlight the need and importance for both the ISPs and the end users of failure prediction, *i.e.*, the proactive detection of DSL line issues. The authors propose NEVERMIND, an active probing-based method that periodically (once a week) probes the state of the users home gateways and based on these measurements predicts whether users will call the helpdesk or not. Predicting failures sufficiently in advance, *i.e.*, several hours or days in advance, allow the ISPs implementing proactive maintenance such that the end-user is not impacted by any failure. The ISP has time to investigate and diagnose in more detail the predicted problem and implement mitigating actions (e.g., pro-actively resyncing the line, uploading a new set of configuration parameters to the DSLAM, gateway reboot).

In this paper, we discuss how to perform online prediction of DSL failures based on *passive* measurements reported by home gateways. Being passive, measurements can be done every few minutes as compared to once every week in NEVERMIND. We compare our prediction results with actual DSL link disconnections reported by the gateways, which is a more precise ground truth label than the helpdesk phone calls in NEVERMIND. Using a longitudinal collection of DSL metrics over 2 years within 98 trial homes, we show that we can predict failures several hours and days in advance with good prediction performance.

2 Problem statement

Our objective is to build a system that does online failure prediction [SLM10], *i.e.*, the system predicts failures ahead of time based on live data provided by home gateways. Our system operates in two phases : (i) model learning and (ii) online failure prediction. During the model learning phase the system learns the failure prediction models. This phase can be performed offline using a training data set and may be repeated periodically to update the models learned. Once the failure prediction models have been learned, we apply the models on live data sent by a set of home gateways. Several types of predictions can be performed with such a system.

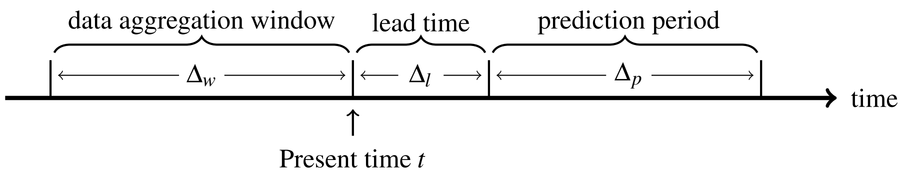

FIGURE 1: Online failure prediction time relations.

The type of predictions covered by this work can be formulated as a classification problem. The predicted class corresponds to the DSL link status ("failure" and "no-failure") within some future prediction period. The time relations are depicted in Figure 1. At present time t, the system predicts some time ahead (lead-time Δ_l) if there will be a failure (or not) within a prediction period of duration Δ_p based on the (aggregated) observations in the data aggregation window of duration Δ_w. The lead-time should be large enough, such that the ISP can implement mitigating actions. Instead of simply providing the predicted class ("failure" and "no-failure"), the prediction model return a failure probability, which is the confidence of the prediction that the predicted class is "failure". More formally, we learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. f takes as input a d-dimensional vector in the feature space $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_d$. The vector is derived from a set of features measured by the home gateway during the data aggregation window, as described in Section 3.1. f returns a failure probability, *i.e.*, a real-valued score between 0 and 1 ($\mathcal{Y} = [0,1]$).

Other types of predictions may be performed, such as estimating the precise time of a failure, or predicting the type or cause of a failure. We leave these as future work.

3 Dataset and failure characteristics

We collected DSL related information from a set of home networks - all subscribers of the same European ISP and spread all over the country - over a period of two years. We instrumented the home gateways of the trial homes with a small client that periodically (once every minute) reports a set of low-level DSL-related parameters to a server. The reported parameters reflect the operational state of DSL link. The number of participating homes varied over the two years, but overall the cumulative set encompass 98 trial homes. The cumulative time of our trial experiment is 35455 days (97 years).

3.1 Features

The features collected in our trial are listed in Table 1. The counters (`cnt`) count the number of related events in a polling period (1 minute). Rates and noise values are the current values as retrieved by our client at polling time. Note that we sample the parameter once at the end of the polling period ; therefore we do not capture potential variations of the parameter during the 1 minute polling period. The client also retrieves the DSL link status, which reports whether the DSL link is up or down. The possible states are : "Up", "NoLink" and "Initializing". We define DSL failures according to this parameter : we consider that there is a DSL failure if this metric is not "Up" (*i.e.*, the status is either "NoLink" and "Initializing"). Finally, we calculate the average and the standard deviation of the above features over the sliding data aggregation

TABLE 1: Features collected for each DSL link.

Feature	Description	Feature	Description
`cntErrSecs`	#secs with CRC err. or LoS	`{up|down}CurrRate`	rate of up/downstream link
`cntSevErrSecs`	#secs with more than 18 CRC errors or one LoS	`{up|down}MaxRate`	theoretical maximum up/downstream link rate according to line attenuation
`cntLoS`	loss of signal (LoS)	`{up|down}NoiseMargin`	noise margin on up/downstream link
`cntLoF`	loss of framing (LoF)	`linkStatus`	DSL link status. Values : Up, NoLink, Initializing
`cntFECErr`	number of FEC errors		
`cntCRCErr`	number of CRC errors		
`cntHECErr`	number of HEC errors		

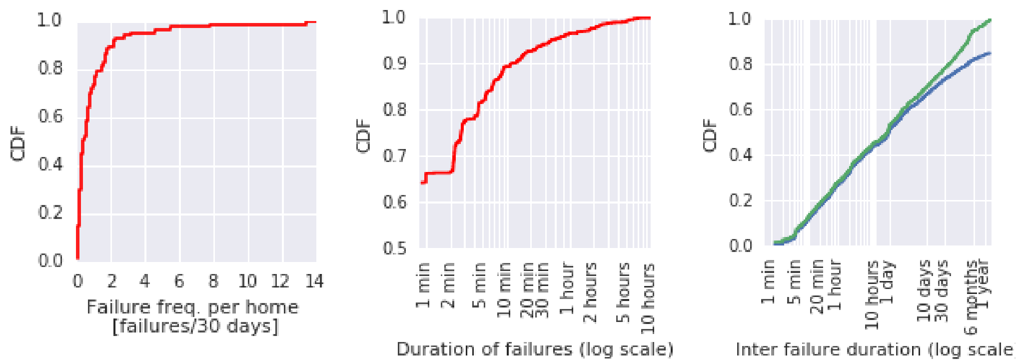

FIGURE 2: Failure characteristics : (left) per-home failure frequency distribution, (middle) failure duration distribution, (right) inter-failure duration distribution.

window (set to a duration of $\Delta_w = 1$ day in our evaluation). In addition, we also include the "time from last failure" that represent the duration since the last failure was observed on the link.

One limitation of our trial setup, is that we do not have more information about the type and cause of a failure - we only know that the DSL link is unavailable. For example, we cannot differentiate failures that are due to a bad conditions on the DSL link, problems on the DSLAM or a cable that has been unplugged. Our prediction methods therefore predicts failures without trying to infer the cause or the type of the failure.

3.2 Failure characteristics

Our dataset contains a total of 1052 DSL failures. Figure 2 (left) plots the distribution of the per-home failure frequency. The mean failure frequency per home is 1.2 failures per month, while the median is 0.5 failures per month. This graph tells us that some homes experience frequent failure, while the majority of the homes rarely fail. The duration of the failures, *i.e.*, the time that the gateway is disconnected, varies. Figure 2 (middle) plots the cumulative distribution function of the observed failure duration. The majority (about 65%) of the failures last less than a minute. Since we use a polling period of 1 minute, we cannot measure the exact failure duration for failures that last less than a minute. About 10% of the failures last more than 10 minutes. Finally, Figure 2 (right) plots the inter failure duration distribution. The figure plots an upper and a lower bound distribution due to left and right-censored failure events (lower bound : set to infinity, upper-bound set to time till measurement end/start). This figure tells us that most failures occur in "bursts". The median inter-failure duration is approximately 1 day.

4 Evaluation of predictive models

Some preprocessing needs to be performed in order to have an unbiased evaluation of the predictive models. First, the proportion of time a DSL link is in failure state is much smaller than the proportion of time the link is up : in our dataset the classes "failure" and "no-failure" are strongly unbalanced. We use the oversampling technique called SMOTE [C$^+$02] to have a balanced dataset. Further, the failures are unevenly spread among the gateways, as we highlighted in Section 3.2. Some gateways fail very often while other fail rarely. In order to remove bias and over-fitting in our evaluation we use *labeled* cross-fold validation (with 5 folds). Labeled cross-fold validation ensures that data of one gateway is not spread over several folds - therefore ensuring that a particular gateway does not boost or penalize prediction performance.

We evaluated a set of classifiers including decision trees, random forests, naive Bayes and AdaBoost. The ROC curve in Figure 3 (left) shows the performance of these classifiers for an data aggregation window of duration $\Delta_w = 1$ day, a lead time $\Delta_l = 6$ hours and a prediction period of duration $\Delta_p = 114$ hours (5 days minus the lead time). Random forests largely outperform the other classifiers. It provides a detection rate (true positive rate) of 44% with a false alarm rate of 1%. The Area Under the Curve (AUC) is 77%. The detection rate has to be interpreted in the light of preventive actions, *i.e.*, in 44% of the cases the ISP has time to investigate and mitigate and reduce the number of failures observed by end-users accordingly.

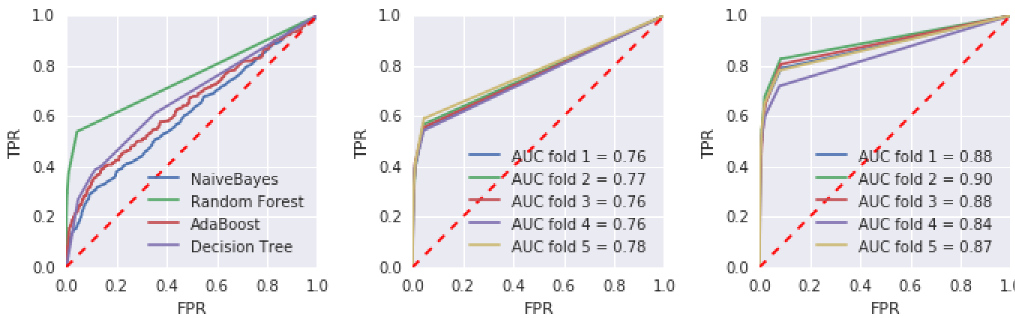

FIGURE 3: ROC curves - false positive rate (FPR) vs true positive rate (TPR) - of the failure prediction models, (left) different classifiers with $\Delta_w = 1$ day, $\Delta_l = 6$ hours, $\Delta_p = 114$ hours, (middle) random forest across different folds with same duration settings, (right) random forest with prediction period as in [J+10] ($\Delta_p = 30$ days, $\Delta_l = 0$)

Further, the ISP can adapt the false alarm rate to the cost of the mitigating action. Typically, the operator might want to use a low false alarm rate for costly mitigation actions. In contrast, the ISP can tolerate high false alarm rate for cheap mitigation actions. Figure 3 (middle) depicts the results of the 5 folds in the same setting for random forest. All folds have very similar curves, *i.e.*, predictions are stable and have no bias.

Figure 3 (right) shows the performance of random forest with prediction times comparable to NEVERMIND, *i.e.*, predicting 30 days in advance without any lead time ($\Delta_w = 0$, $\Delta_p = 30$ days). At a false alarm rate of 1% we have a detection rate of 56%. We achieve an accuracy of 82%. It is difficult to compare this with the results of NEVERMIND, since NEVERMIND only evaluates a subset (top 20K) of positive failure predictions on which they achieve an accuracy of 40%.

To determine which features impact the most the classification results we calculate the Gini importance, as shown in Figure 4. The features `timeFromLastFailure`, averages of `downMaxRate`, `downNoiseMargin`, `upMaxRate` and `upNoiseMargin` a are the most discriminant. Non-disciminant features include `cntFECErr`, `cntLoS`, `cntLoF`.

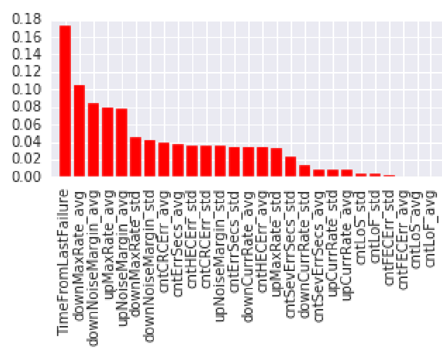

FIGURE 4: Feature importance in random forest.

5 Outlook

We envision several future works. First, we would like to study of the root cause of DSL failures and the possible mitigating actions. Second, the type of prediction method discussed in this paper only determines if there will be a failure soon. It however does not try to predict when precisely the failure will happen. In future work, we would like to determine more precisely when the failure will happen. Typically we would like to continuously estimate for each gateway the time-to-failure. We plan to use survival analysis, that is a specific type of regression model where a set of observations can be censored.

Références

[C+02] Nitesh V. Chawla et al. SMOTE : synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 2002.

[J+10] Yu Jin et al. NEVERMIND, the problem is already fixed : proactively detecting and troubleshooting customer DSL problems. In *Conext*, 2010.

[SLM10] Felix Salfner, Maren Lenk, and Miroslaw Malek. A survey of online failure prediction methods. *ACM Computing Surveys*, 42(3) :1–42, 2010.