Fault Diagnosis in DSL Networks using Support Vector Machines

A. K. Marnerides[a,*], S. Malinowski[b], R. Morla[c], H. S. Kim[d]

[a]*School of Computing & Mathematical Sciences, Liverpool John Moores University, Liverpool, UK*
[b]*AgroCampus Ouest, IRISA, Rennes, France*
[c]*INESC Porto, Faculty of Engineering, University of Porto, Portugal*
[d]*Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, USA*

Abstract

The adequate operation for a number of service distribution networks relies on the effective maintenance and fault management of their underlay DSL infrastructure. Thus, new tools are required in order to adequately monitor and further diagnose anomalies that other segments of the DSL network cannot identify due to the pragmatic issues raised by hardware or software misconfigurations. In this work we present a fundamentally new approach for classifying known DSL-level anomalies by exploiting the properties of novelty detection via the employment of one-class Support Vector Machines (SVMs). By virtue of the imbalance residing in the training samples that consequently lead to problematic prediction outcomes when used within two-class formulations, we adopt the properties of one-class classification and construct models for independently identifying and classifying a single type of a DSL-level anomaly. Given the fact that the greater number of the installed Digital Subscriber Line Access Multiplexers (DSLAMs) within the DSL network of a large European ISP were misconfigured, thus unable to accurately flag anomalous events, we utilize as inference solutions the models derived by the one-class SVM formulations built by the known labels as flagged by the much smaller number of correctly configured DSLAMs in the same network in order to aid the classification aspect against the monitored unlabelled events. By reaching an average over 95% on a number of classification accuracy metrics such as precision, recall and F-score we show that one-class SVM classifiers overcome the biased classification outcomes achieved by the traditional two-class formulations and that they may constitute as viable and promising components within the design of future network fault management strategies. In addition, we demonstrate their superiority over commonly used two-class machine learning approaches such as Decision Trees and Bayesian Networks that has been used in the same context within past solutions.

Keywords: Network management, Support Vector Machines, supervised learning, one-class classifiers, DSL anomalies

1. Introduction

The incremental and demanding user access in various types of networks has enforced domains such as network management and infrastructure maintenance to evolve as crucial and highly challenging engineering tasks for network operators. Undoubtedly, network management is acknowledged as a domain of critical importance and it has received a considerable level of attention by the networking research community. As a core element for traffic engineering (i.e. TE), network management holds several objectives to confront under the scope to achieve an optimal and adequate operation of a network(s). Naturally, such objectives are concerned with several aspects of a networked environment like fault/anomaly detection, traffic classification, performance, security, configuration management as well as planning, designing and administration of the networked infrastructure(s). Clearly, all the aforementioned aspects are inter-dependent and a generic network management methodology should closely consider them up to a great scale when firstly designed.

Current best practices comply with older methodologies and engage Expert System (ES) approaches that unavoidably involve at the initial stage an exhaustive manual inspection of network/application layer traffic traces (e.g. SNMP, NetFlow records) and maintenance logfiles (e.g. syslog) under the intention of characterizing normal operation and extracting abnormal patterns [1–3]. Nonetheless, such techniques blindly rely on the empirical observational analysis of an experienced operator and they can never be fully entrusted.

Apart from the operator inspection, ES-based network management methodologies as proposed in past and current literature [1, 2, 24–26] rely traditionally on learning capabilities invoked by various Machine Learning (ML) algorithms [1–4, 13]. Learning, as adopted by the domains of machine learning (ML) and artificial intelligence(AI), holds a critical role within network management schemes since its sole purpose is to provide a level of proactive knowledge regarding the normal and the likely abnormal operation that possibly resides in a networked environment. There is a plethora of network or system-wise features (e.g. packet records, router CPU utilization statistics, etc.) [5, 6] that can be used throughout a learning phase and in parallel a given management task (e.g. anomaly detection) might require an explicit type of features (e.g. packet features

[*] Corresponding author
Email addresses: `a.marnerides@ljmu.ac.uk` (A. K. Marnerides), `simon.malinowski@gmail.com` (S. Malinowski), `r.morla@fe.up.pt` (R. Morla), `kim@ece.cmu.edu` (H. S. Kim)

such as src/dst IP addresses) [17].

Nevertheless, the vast majority of many ML-based schemes that address an explicit network management task (e.g. traffic classification, anomaly detection), have been formulated under particular algorithms that either compose a supervised [13, 14, 19, 22, 23, 34], semi-supervised [14, 17, 20] or even unsupervised approach [12, 17, 21]. Given the experimental results presented by many studies, every algorithm initiates its own advantages and disadvantages with respect to the level of *accuracy* accommodated when employed. In this paper we aim at presenting the benefits offered by a particular type of learning that is derived by Support Vector Machines(SVMs) under a particular network management scheme dedicated for fault diagnosis in DSL-based networks. We following describe the exact problem statement and further state our contributions.

1.1. Problem Description & Motivation

The problem addressed herein spans mainly into two main domains that are inter-related; (a) network management in DSL-based networks and (b) fault diagnosis under machine learning using SVMs. Given the fact that various types of ES-based approaches are heavily dependent on the empirical observation of experienced network operators [1, 2, 24, 25] we initially want to propose a new scheme based on SVMs that would require their minimal interference on empirically inferring faults. Thus, our first problem is related with the effort at providing a sufficient learning methodology for the explicit task of identifying DSL-level faults.

In parallel, this study targets to assess fault diagnosis under the real conditions existing on a DSL infrastructure. Unfortunately, such infrastructures as deployed in the majority of service providers are minimally equipped with fault characterization capabilities on the installed Digital Subscriber Line Access Multiplexers (i.e. DSLAMs). In particular and as reported in [7, 14, 37], there are many cases where due to improper configuration, deployed DSLAMs in real infrastructures do not correctly flag a detected DSL anomaly such as a signal degradation or a power cut as it happens with the dataset that we use in this work. Consequently, this lack of adequately categorizing and reporting faults to the centralized management infrastructure administered by an operator leads to inaccurate interpretation of faults/anomalies occurring on the upper layer of the actual service distribution network (e.g. IPTV). With no doubt, several degrading service-related events are directly linked with the performance of the installed DSLAMs [7, 14], thus a thorough understanding of DSL-level anomalies is highly essential. In parallel, a characterization of DSL-level anomalies under the non-pragmatic assumption that all DSLAMs are fault and anomaly-aware as it happens in all the proposed methodologies as in [5, 6, 37–39] leads to flawed and inaccurate conclusions.

Apart from the pure fault management aspect in our problem, this work thoroughly presents and discusses the outcomes of SVMs on classifying DSL-level anomalies. As already mentioned, the greatest amount of the deployed DSLAMs in our datasets were considered as non-anomaly detection aware (i.e. non-AD) and the largest number of anomalies were flagged in a default manner without the root cause of failure being specified nor classified. On the other hand the anomaly-aware DSLAMs (i.e. AD-aware) were in a position to correctly flag all the events into two major anomalies: signal degradations and power cuts. However the number of signal degradations was much greater than the power cuts, thus in our two-class SVM classification (described in section 5) we have experienced an issue known as *data imbalance*. In general, a dataset is imbalanced if the classification categories are not approximately equally represented. Data imbalance is a known problem within the ML literature and it causes a high risk of inaccurately constructing SVM models for the known anomalies. In simple terms, the imbalanced nature of such training samples would invoke high classification errors and problematic labeling of the training instances that naturally engage high rates of misclassification throughout the testing phase [10].

Moreover, due to the tremendously smaller number of AD-aware DSLAMs in comparison with non-AD DSLAMs in the same network we also aimed at using the classification outcomes resulted by the AD-aware measurements as inference solutions in order to label the unclassified events flagged by the non-AD DSLAMs.

1.2. Contributions

We explicitly address the real conditions undertaken on a DSL infrastructure and introduce an offline formulation for fault-diagnosis for serving the demanding domain of network management. Based upon the problem description provided earlier we suggest a learning framework that can sufficiently classify anomalous events present on the DSL infrastructure of a large European ISP. We following highlight our contributions.

- Due to the *data imbalance* experienced in our datasets we initially migrate from the traditional supervised SVMs and propose a semi-supervised scheme in order to infer the events captured from the non-AD DSLAMs. We show that given this data-specific problem we could have opposing outcomes regarding the real nature of the events from the non-AD DSLAMs.

- By virtue of the learning bias initiated by the *data imbalance* during the learning phase we go a step beyond and propose the applicability of one-class SVMs as an opposing solution to the semi-supervised SVM scheme. To the best of our knowledge, this formulation has never been used in the context of network management and we argue in fair of its accuracy and applicability particularly in the case where imbalanced datasets are present.

- We show that the resulting one-class SVM formulations reach more than a 95% of overall accuracy on identifying a single type of anomaly. Moreover we illustrate that a collaborative employment of one-class SVM models that are able to recognize a single type of event can sufficiently infer the events captured by the non-AD DSLAMs.

- We provide a coherent discussion and comparison between the two-class and one-class SVMs either on a semi-supervised or supervised nature. Our interpretation on the generated outcomes hold as a basis for the manifestation of SVM-based learning approaches for ES systems in any type of a networked infrastructure.

- We show that the developed one-class SVM classifiers demonstrate higher accuracy with respect to the precision, recall and the F-score classification performance metrics than the traditionally used Decision Tree (DT) and Bayesian Network (BN) algorithms as used in [38] and [39] and they overcome the aspect of imbalanced datasets as it happens in most of two-class supervised algorithms used in the context of fault management.

The rest of this paper is structured as follows: section 2 provides a brief background on related work, section 3 presents the data used within our experimentation as well as the metrics used for measuring the classification performance obtained in our work. Section 4 demonstrates a statistical characterization of the two known anomalies and further exhibits how this work composed the feature sets used within the classification process. Subsequently, section 5 discusses the outcomes of the supervised and semi-supervised two-class SVM classification performed in our datasets and Section 6 describes the evaluation with respect to the construction of our one-class SVM as well as a comparison with the DT and BN models. Section 7 demonstrates the evaluation of the resulted one-class models and their comparison with the DT and BN models under testing unlabeled SyncTrap events as captured from the non-AD aware DSLAMs whereas section 8 compares the results between the two-class and one-class SVMs. Finally, section 9 summarizes and concludes this paper.

2. Related Work

Fault management and troubleshooting is considered as an integral component for network management and is decomposed into three main sub-domains; fault detection, fault localization and testing [1–4, 7, 14]. Fault detection refers to the act of capturing indications (e.g. SyncTrap events [7, 14] regarding the anomalous behaviour of networked environment whereas localization is considered as the process of analyzing such indications under a mathematical framework [5, 25, 37]. Testing is the procedure for determining the precise root cause of a failure based on the captured indications [25].

According to [25] and Jin et. al. in [37], fault management and troubleshooting methods place an effort to include all three of the aforementioned sub-domains. Given the studies in [1, 24–26] this ability of merging fault detection, localization and testing in one single mechanism is mainly manifested by ML-based techniques. Studies as in [27, 30–32] propose ES-based frameworks where the ultimate decisions regarding the localization and testing of faults are derived by inference engines formulated by ML techniques such as DT and BN. In fact, for the particular objective of characterizing DSL-level anomalies as demonstrated in this work, there has been a number of techniques that mainly aimed to address this problem using machine learning approaches as in [37–39]. However, to the best to our knowledge, none of the proposed ML-based schemes have not employed Support Vector Machines (SVMs) that have shown significant accuracy in other disciplines such as anomaly detection [12] and Internet traffic classification [13, 19].

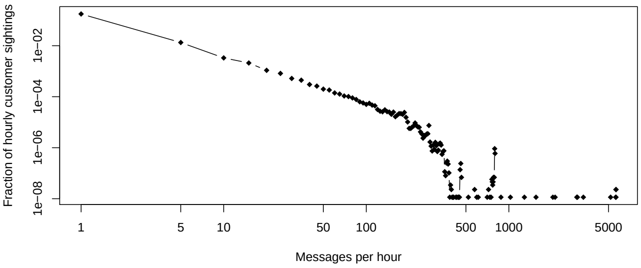

Figure 1: Distribution of customer message counts per hour with regard to hourly customer sightings, normalized to the total number of hourly customer sightings in the 720 time bins in the month.

Moreover, the detection part in all of these proposals was still heavily dependent on the empirical observation by a network operator. According to [25] these propositions indicated that such schemes hold several drawbacks with respect to the training phase as well as with the accuracy rates obtained by those particular ML algorithms. In addition, given the supervised nature of these techniques, there is little space of achieving the detection of new faults particularly for large and complex networks.

3. Data Description, Labeling & Classification Metrics

3.1. Raw data

As also described in [7, 14], the dataset we use in this paper has been provided by a major European commercial IPTV service provider[1]. We use DSL logs containing a SyncTrap message for each time the DSL connection to any customer is lost or re-established[2]. Our analysis is based on one hour time bins and we keep track of customer sightings and customer SyncTrap message counts in each hour. Figure 1 shows the distribution of customer message counts per hour as being normalized to the total number of hourly customer sightings in the 720 time bins of the month. The presented distribution is linear in the log-log scale of the graph approximately below 200 messages per hour. Such distribution shape indicates a power-law relationship between hourly customer sightings and customer message count per hour, in which many customers generate small number of messages per hour and a small number of customers generate a large number of messages per hour. Naturally, the latter fact may confirm the intuition that most of the DSL customers have normal behavior (i.e. small number of messages per hour) and only a smaller number of customers have anomalous behavior.

[1] Due to privacy and business concerns we do not state the name of the provider.

[2] We clarify that in this work we are strictly concerned with the DSL-level faults and we do not intent at showcasing a correlation of DSL-level faults with IPTV application-specific characteristics since our aim is to provide a holistic fault classification technique for anomalies of the DSL-level infrastructure regardless of the upper layer service that is supported.

Nonetheless, in the rest of this paper we focus on the set of high message rate customers. We consider only pairs of (customer c, hour h) observations for which customer c has more than 20 SyncTrap messages in an hour h. For each of these pairs, we process a time series composed of the inter-arrival times between two consecutive SyncTrap messages related to c that occurred during hour h.

Table 1: Datasets used - Labels : A = Signal Degradations, B = Power Cuts

$Dataset$	$No.ofRecords$	$Anomaly/Label$
$\mathcal{S}$	$448,544$	A, B and unlabelled
$\mathcal{L}$	$135,793$	A
$\mathcal{T}$	3300	B
$\mathcal{U}$	$309,451$	unlabelled
Ω	$53,300$	A, B

3.2. Data Labeling

A number of DSLAMs in the deployed network are configured to notify a connection problem either due to signal degradations (type A) or power cuts (type B). Under a connection problem scenario, the generated SyncTrap message is correspondingly labeled by the AD-aware DSLAM. We have labeled the time-series for which all the events were of the same type (A or B) and the whole dataset is contained within the set $\mathcal{S}$. Our labeling splits $\mathcal{S}$ into three subsets $\mathcal{L}$, $\mathcal{T}$ and $\mathcal{U}$. $\mathcal{L}$ is composed by all the vectors that are labeled with A and $\mathcal{T}$ with those labeled with B, where $\mathcal{U}$ defines the set of all the unlabelled vectors. In order to have a mixed labelled dataset for testing purposes, we also define a subset Ω that is resulted by mixing a subset of $\mathcal{L}$ with the full dataset residing in $\mathcal{T}$. The Ω subset was mainly used for the validation of our two-class SVMs since it aided at reasonably reducing the high imbalance between the type-A and the type-B labeled records. However, despite this effort, the initial labelled record sets still contained an unbalanced nature that had to further be algorithmically confronted by the employed SVM formulations.

As evidenced by table 1 the dataset $\mathcal{S}$ consists of a total of $448,544$ records that represent the total number of events captured in the period of one month from both the AD-aware and non-AD aware DSLAMs. Thus, the records in $\mathcal{S}$ contain type-A (signal degradations), type-B (power cuts) and unlabeled events. The exact number of reported signal degradation and power-cut events resides in the datasets $\mathcal{L}$ and $\mathcal{T}$ respectively. In practice, both $\mathcal{L}$ and $\mathcal{T}$ were captured from the AD-aware DSLAMs. On the other hand, the dataset $\mathcal{U}$ denotes all the events captured from non-AD aware DSLAMs and it can be easily observed that they compose the larger subset within the overall trace represented by $\mathcal{S}$.

3.3. Classification Performance Metrics

In order to measure the accuracy performance of our SVM schemes as well as the Decision Tree and Bayesian Network approaches, we adopt the classification metrics of *accuracy, recall, precision, F-score* and *G-mean*. As shown from their formulation denoted below, these metrics make use of the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$G - mean = \sqrt{\frac{TP \times TN}{(TP + FN) \times (TN + FP)}}$$

In addition, the process of validating the performance of our composed classifiers follows a *k-fold* approach [8]. In particular we employ a 10-fold cross-validation process that iteratively selects random segments of data in order to examine the accuracy rate of a training model [8].

4. Feature Analysis & Composition

4.1. Examining Initial Statistical Features

As already mentioned, the simplistic inter-arrival timeseries for each client seemed inadequate with respect to significantly aid the discrimination of the DSL-level anomalies that occur. However, it was necessary to validate this speculation. In particular, our initial investigation targeted at verifying on whether the empirical distributions of the inter-arrival series of a given client that indicates a signal degradation or a power cut had similar characteristics or otherwise. Therefore we chose to employ the commonly used two-sample Kolmogorov-Smirnov test [35] (i.e. K-S test) while comparing such distributions. The two-sample K-S test is a non-parametric approach that allows to identify on whether two one-dimensional empirical distributions differ.

Let $T_s = t_1, \ldots, t_m$ and $T_p = t_1, \ldots, t_n$ represent the inter-arrival time series of two clients (i.e. two different DSLAM paths) in an hour h where T_s is a series composed by consequent signal degradation SyncTrap events on a given client and T_p the series for a client that experienced a power cut. In addition, T_s of size m has the cumulative distribution function (i.e. c.d.f) $F(x)$ and T_p of size n a c.d.f with $G(x)$ and their corresponding *empirical* c.d.fs as $F_m(x)$ and $G_n(x)$ respectively. Under these terms, the K-S test holds two hypotheses, the null hypothesis $H_0 : F = G$ and the rejection of the null hypothesis $H_1 : F \neq G$. In order to validate the null hypothesis via measuring the statistical (in)significance between $F_m(x)$ and $G_n(x)$ it is required to compute the Kolmogorov-Smirnov statistic D_{mn} defined as:

$$D_{mn} = \left(\frac{mn}{m+n}\right)^{1/2} \sup_{n} |F_m(x) - G_n(x)| \quad (1)$$

The null hypothesis is rejected in the statistical significance

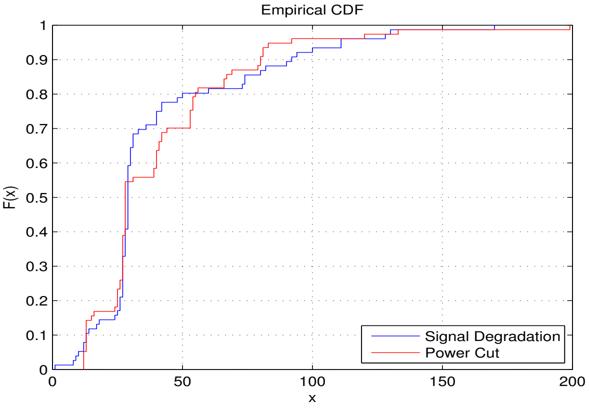

Figure 2: Empirical CDFs for two different clients that experienced loss of service due to different faults within the same time bin.

level a if

$$\sqrt{\frac{mn}{m+n}}D_{mn} > K_{\alpha} \tag{2}$$

where K_{α} of statistical significance level α can be found from the relationship of the Kolmogorov distribution K denoted as:

$$K = \sup_{t\in[0,1]} |B(t)| \tag{3}$$

and has the following relationship [3]

$$Pr\,(K \leq K_{\alpha}) = 1 - \alpha \tag{4}$$

The statistical significance level α is the most critical parameter that actually determines the sensitivity at which the K-S test will reject the null hypothesis or otherwise. Thus, after experimentation we kept it to hold the value of 0.05 since tuning this parameter would provide a pre-defined and biased result in favor of the null hypothesis that we wish to prove.

Nevertheless, given equations 1 and 2 we have employed the K-S test using a dedicated function in almost all the clients that experienced power cuts and compared their corresponding time series samples with clients that suffered from signal degradations. The incentive behind these tests was to verify that these particular anomalies do not greatly differ from a statistical viewpoint and that inference techniques derived by ML are necessary in order to differentiate them[4]. Indeed, throughout this verification stage we have witnessed that more than 93% of the power-cut timeseries whilst compared with timeseries composed by signal degradation events have had high levels of similarities, thus complying with the null hypothesis of the K-S test. For the purpose of this paper we provide Fig. 2 in order to illustrate how similar were the resulting CDFs between the two different types of anomalies.

4.2. Feature Description

Given the SyncTrap inter-arrival timeseries for each client as described earlier (section 3) we have extracted 10 statistical features in order to construct the SVM-based as well as the Decision Tree and Bayesian Network models discussed and compared in following sections. In particular the meta-features from the timeseries are the following:

- mean and variance of the timeseries
- number of elements of the timeseries
- normalized Shannon entropy for each client timeseries
- 6 parameters related to Hidden Markov modeling of the timeseries

On the contrary with the mean, variance and the number of elements invoked by each inter-arrival timeseries, the formulation of the normalized entropy and HMMs were non-trivial. Therefore we briefly describe next why and how they were used within our experiments.

4.3. Shannon Entropy

The Shannon entropy has shown substantial results and it has been established as a robust and reliable metric to use in the area of anomaly detection [17, 18, 28, 29]. We compute the Shannon entropy of a timeseries under a histogram-based technique.

Let $T = t_1, \ldots, t_n$ represent the per-client event timeseries. We proceed with an estimation of the Shannon entropy based on an histogram whose bin width complies with the Freedman-Diaconis rule [11]. According to this rule, the width of the bins for a time-series T of length n is

$$w = 2 \times IQR(T) \times n^{-1/3}, \tag{5}$$

where $IQR(T)$ is the interquartile range of T. The produced histogram allows modelling T as a realization of a discrete random variable whose possible values $x_1, \ldots, x_w$ are defined by its bins. The probabilities $p_1, \ldots, p_w$ associated with these values are computed directly from T and the bins. According to this modelling, the Shannon entropy of T is estimated as

$$H(T) = -\sum_{i=1}^{w} p_i \log p_i, \tag{6}$$

Since the range of the entropy depends on the number of bins w, we normalize $H(T)$ by $\log_2(w)$, that is the maximum value of the entropy of a discrete random variable with w symbols.

The incentive behind the usage of this particular feature lies with the promising and justifiable results obtained by past research [7, 17, 18] as well as by the promising findings during our analysis. We have specifically examined the distributional characteristics of the Shannon entropy values in our whole dataset. As presented in our earlier work in [7], the Shannon entropy values derived by the SyncTrap inter-arrival series for each client have pinpointed several anomalous peaks that could have not been spotted with simple statistical features

[3] $B(t)$ in equation 3 denotes the Brownian bridge of a continuous stochastic process.

[4] Within our experimentation we have selected to compare time series of a relatively same size since a great difference with respect to sample size would produce biased results.

Table 2: Top ten anomalous hourly bins from the total of 720 hours (one month) as indicated by Shannon entropy-based and simple SyncTrap event-based statistics. (H(T) = Shannon Entropy, IA = Inter-Arrival Time)

FEATURE	*PEAKHOURS*
H(T) Mean	1,388,3,383,17,380,548,365,367,15
H(T) Var.	657,608,241,259,653,714,417,630,86,659
Mean Event IA	385,388,382,548,1,3,400,378,367,393
No_Events Mean	490,62,582,702,293,486,484,273,482,577
Event IA Var.	497,228,369,16,371,402,390,569,548,540
No_Events Var.	490,61,190,402,210,9,13,273,296,293

(e.g. inter-arrivals, SyncTrap event volume per client). In particular, our work in [7] demonstrated that the investigation of per-hour mean and variance values of the Shannon entropy as derived by the inter-arrival timeseries of each individual client can map types of anomalies within particular values.

By contrast with the simple per client inter-arrival timeseries, the two types of anomalous events (i.e. signal degradations, power cuts) corresponded within particular entropy values. For instance, high Shannon entropy variance in the majority of cases pointed large numbers of signal degradations in a given hourly bin whereas mid values showed power cuts instead. Furthermore, high volume size series (i.e. when a client experienced a large number of SyncTrap events regarding a specific failure) reported by non-AD DSLAMs were not obtaining extremely large variance or mean entropy values.

In general the computed Shannon entropy values had the advantage at providing a more descriptive metric for anomalous hourly peaks opposing the least accurate simple metrics. As depicted in table 2 the mean and variance Shannon entropy values could include the majority of anomalous hours indicated by simple metrics but they could denote which of those were in reality important for an operator to consider. As described in [7], a manual inspection undertaken in order to verify the outcomes of the Shannon entropy computations, justified that this particular metric on a per-client basis is surely a good option to include within a more advanced inference task. Thus, our ML-based approach utilized the Shannon entropy values for each client as one of the discriminative features within the construction of the training sets in both the two and one-class SVM methodologies.

4.4. Hidden Markov Modeling

Hidden Markov Models (HMMs) are widely used to describe complex probability distributions in time series and are well adapted to model time dependencies in such series. We briefly recall in this section how HMMs are defined. Any interested reader may find more details about HMM modelling in our earlier work in [14] as well as the Rabiner work in [36]. Nevertheless, an n-state HMM is composed of:

- A set of states $\mathcal{S} = \{S_1, \ldots, S_m\}$,
- A state transition matrix $\Gamma = (\gamma_{i,j})_{1 \leq i,j \leq m}$
- The probability distributions $\mathcal{B} = \{b_1, \ldots, b_m\}$ of the observations associated with the set of states
- An initial probability distribution $\Pi = \{\pi_1, \ldots, \pi_m\}$

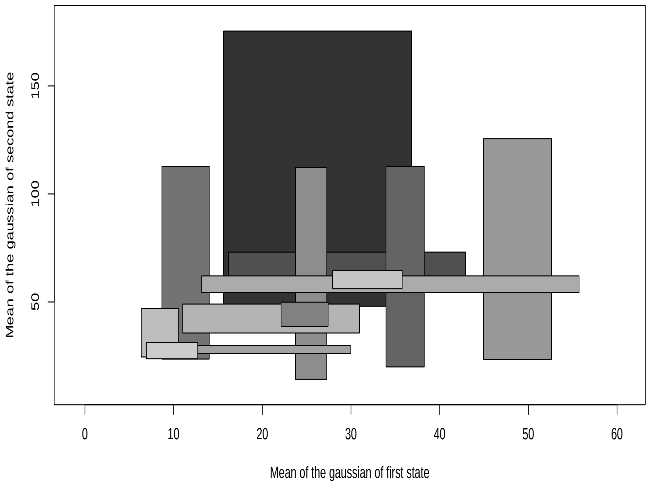

Figure 3: Representation of the 13 clusters obtained in the 6-dimensional HMM parameter space.

The set $\Lambda = (\Gamma, \mathcal{B}, \Pi)$ completely defines a HMM. For a given time instant t, the element $(\gamma_{i,j})$ of Γ represents the probability that, at time $t+1$, the model is in state j, given that it was in state i at time t. In other words, assuming that q_t denotes the value of the state of the model at time t,

$$\gamma_{i,j} = \mathbb{P}(q_{t+1} = S_j \,|\, q_t = S_i). \tag{7}$$

Observations are generated by an HMM depending on the current state. For an observation Δ_t and a state index i, $b_i(\Delta_t) = \mathbb{P}(\Delta_t \,|\, q_t = S_i)$. Note that the distributions b_i can be either discrete or continuous. Finally, π_i, $1 \leq i \leq m$ represents the probability that the initial state q_1 is equal to S_i. In other words, $\pi_i = \mathbb{P}(q_1 = S_i)$.

Hence, an HMM represents a distribution probability parameterized by Λ. For an HMM h represented by the set of parameters λ_h it is possible to compute the likelihood $\mathcal{L}(\Delta|\lambda_h)$ of an observation vector Δ given h. This value is related to the probability that Δ has been generated by h. We can then fit an HMM to the time series Δ. This step is one of the three main issues when dealing with HMMs (referenced as Problem 3 in [36]). This fitting is realized using a maximum-likelihood algorithm. The set of parameters λ_Δ of the HMM fitted to Δ is selected so that it maximizes the likelihood of Δ given λ_Δ. Hence,

$$\lambda_\Delta = \arg\max_\lambda \mathcal{L}(\Delta|\lambda). \tag{8}$$

This is a classical problem in parameter estimation that can be solved using the Baum-Welch algorithm [15, 16]. This algorithm is iterative and ensures the increase of the likelihood of the observation given the set of parameters at each iteration, until it converges to a local maximum of the likelihood function. For each DSL connection-hour sighting s with a large number of SyncTrap events we compute the HMM parameter set λ_Δ^s. For simplicity and as each s only has one Δ^s, we rewrite the HMM parameter set as λ^s.

In this work, we use the Baum-Welch algorithm to learn 6 parameters of two-state 1-dimensional Gaussian HMMs of the inter-arrival SyncTrap series of each DSL client: 1) 4 parameters for the mean and variance of the Gaussian that represents the observation distribution of each state; and 2) two parameters for the transition probabilities between the two states. Nevertheless, interpreting client representations in the 6-dimensional HMM parameter space is not straightforward. Therefore, we have clustered the 6-dimensional representation of all client time series using the hierarchical clustering method with dynamical cutting proposed in [42] and resulted in the 13 clusters represented in figure 3. Each box in figure 3 represents a cluster whereas the center of the box represents the mean of the observation distribution for the first and the second state of the cluster. Box widths and heights represent the variance of the observation distribution for each state of the cluster. The grayscale box represents the temporal structure of the event series. In particular, it encodes the probability p of changing state between two consecutive samples, $p = \mathbb{P}(S_t \neq S_{t+1}) = \gamma_{1,2}\mathbb{P}(S_t = 1) + \gamma_{2,1}\mathbb{P}(S_t = 2)$. Finally, a lighter box represents higher probability p of changing state.

Overall, figure 3 shows clusters with higher state changing probability p (lighter boxes) at both high and low averages, as well as clusters with lower p at both high and low averages. Consequently this outcome demonstrates that the temporal structure of the data is not aligned with a simpler metric like the average. In addition, it illustrates the value of HMMs in distinguishing client timeseries that would be grouped together when compared using simple metrics. For example, we could think about using a simpler version of a two-state HMM without temporal information – which is equivalent to a two-state Gaussian mixture. In this case and for several boxes that are geometrically close in figure 3 yet have different gray levels, we could say the HMM model is able to distinguish timeseries that have a similar geometrical representation and that a simpler model like the Gaussian mixture could not.

5. Two-Class SVM models for DSL-level Anomalies

5.1. Two-class SVM formulation

Our experimentation is conducted under exploiting two-class kernel-based SVMs. In general, the main objective of a binary SVM is to build a hyperplane with maximum distances to the nearest point of each class. Kernel-based SVMs explicitly enable to search for non-linear separation between the classes, thanks to the kernel trick in [40]. Thus, let our training set be composed of l normalized instance-label pairs such as (x_i, y_i), $i = 1, \cdots, l$ where each pair has n inputs, thus $x_i \in R^n$. Also, the labeling from y_i is binary, hence $y \in \{-1, +1\}$. According to the original definition provided in [33] and explained in [8], the SVM algorithm aims to solve the following optimization problem:

$$\min_{w,\beta,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{9}$$

subject to

$$y_i(w^T \phi(x_i) + \beta) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \cdots, l \tag{10}$$

The function $\phi(x_i)$ is a mapping function that maps the original data points to a higher dimensional space and $C > 0$ is an SVM-specific regularization parameter. Due to the high dimensionality invoked by the weight vector variable w the SVM solves the dual problem using the ρ_i Lagrange multipliers:

$$\min_{\rho} \frac{1}{2} \rho^T U \rho - \epsilon^T \rho \tag{11}$$

subject to

$$y^T \rho = 0, 0 \leq \rho_i \leq C, i = 1, \cdots, l \tag{12}$$

where $\epsilon = [1, \cdots, 1]^T$ is the vector that contains all the positive labels and U is an lxl semidefinite matrix that can be expressed as:

$$U_{i,j} = y_i y_j K(x_i, x_j) \tag{13}$$

In our case, the function $K(x_i, x_j)$ is a Radial Basis Function (RBF) kernel and is defined as:

$$K(x_i, x_j) = exp\Big(- \gamma \parallel x_i - x_j \parallel^2 \Big), \gamma > 0 \tag{14}$$

In summary, at the point where the problem of equations 11 and 12 is solved and under the dual-prime relationship then the optimal q satisfies:

$$w = \sum_{i=1}^{l} y_i \rho_i \phi(x_i) \tag{15}$$

Hence, a resulted classification decision function is feasible as follows:

$$sgn\Big(w^T \phi(x) + \beta\Big) = sgn\Big(\sum_{i=1}^{l} y_i \rho_i K(x_i, x) + \beta\Big) \tag{16}$$

Given equation 16 a subsequent step is to gather the corresponding and best-fit support vectors (SVs) alongside all trained weight-label instances (i.e. (y_i,ρ_i)) which in practise provide the means to predict the label of a newly inserted x_i.

5.2. Evaluating Two-Class Supervised SVMs

At first, we build a traditional two-class SVM based on the whole set Ω (i.e., highly imbalanced) that is labeled with type-A (i.e. signal degradation anomalies) or with type-B (i.e. power cut anomalies) and is composed of 53,300 10-dimensional vectors where 3,300 of these vectors are labeled as type-B anomalies. Anomalies A and B are extremely imbalanced in our case, as only 6.6% of the set is labeled with type B whereas the remaining with type A. By default, in the case of an imbalanced dataset, the two-class SVM is biased, as the majority class (i.e. type-A) tends to push the decision boundary towards the minority class [10]. Hence the training phase is experiencing the data imbalance problem that we have clearly stated in the introduction of this work. Consequently, this effect leads the classifier to produce an estimated model that tends to allocate testing records towards the majority class label. Nonetheless, Ω is first split into a training set and a testing with an analogy with respect to Ω of 70%-30%[5] respectively for each since such

[5] Throughout all the tests reported herein, we utilised a range of analogies between the testing and training samples but given the imbalanced nature of the initial dataset we found that 70% for training and 30% for testing was giving the most representative results while validating our classiffiers.

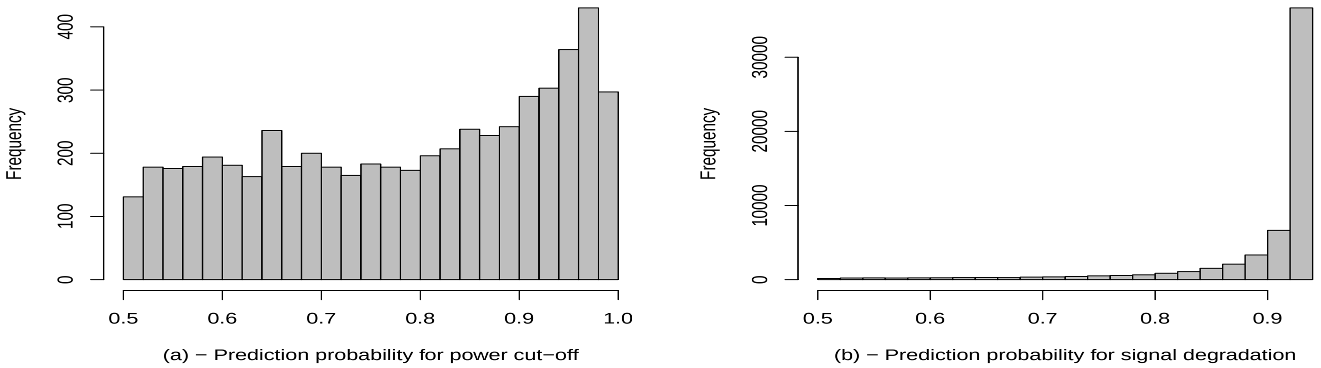

Figure 4: Histogram counts of prediction probabilities when the unlabeled data set is predicted with SVM_1.

Table 3: Classification performance in terms of confusion matrix obtained with SVM_1.

		True label	
		A	*B*
Predicted label	*A*	14,959	727
	B	58	256

Table 4: Classification performance in terms of confusion matrix obtained with GSVM-RU (SVM_2).

		True label	
		A	*B*
Predicted label	*A*	11,340	243
	B	3,677	740

percentages align with robust training as indicated in [20]. In our case the testing set is composed of 16,000 samples where 983 labeled as type-*B* anomalies. Initially, a two-class SVM is trained using the training set with tuned parameters under a 10-fold cross validation technique. Subsequently, the testing samples of the testing set are predicted using the learned model. The generated confusion matrix derived by these predictions is given in Table 3.

Table 3 as well as Fig. 4 provide a piece of evidence regarding the impact of the high number of labels *A* with respect to the classification outcome. As anticipated, the greatest majority of samples is predicted as type-*A* anomalies by the generated SVM_1 model. Despite the fact that the overall accuracy performance for the SVM_1 model is quite high by reaching a 95%, it is not considered as convincing since we deal with an imbalanced dataset. Complementary to the overall accuracy rate, the precision performance for both labels *A* and *B* is good, whereas the recall measure for *B* is moderate (i.e. 26%). This low recall value is due to the high number of real type-*B* records that are predicted as type-*A*. At the same time, the G-mean associated with these predictions attained a 51% indicates a flawed prediction process with respect to *sensitivity* (i.e. recall) and *specificity*[6]. Simply enough and as evidenced by Fig.4 this low G-mean rate denotes that the SVM_1 model had an extremely low probability at retrieving a related training record when examining a given testing record (i.e. sensitivity - recall) and that the level of confidence at precisely matching a testing with a training record was similarly low (i.e. specificity).

In order to improve the G-mean metric our investigation experimented with two methods. The first method randomly under-samples the type-*A* class and builds a model that relies on an improved balanced training set, whereas the second technique selects the most informative samples from the majority class (i.e. type-*A* classes) and further constructs a balanced data set. In particular, the latter technique is called GSVM-RU and it was firstly presented in [10]. By considering all the resulted outcomes, both methods lead to similar results in terms of G-mean improvement, thus we choose at presenting the confusion matrix generated by the GSVM-RU technique. This confusion matrix is given by Table 4 and shows the best classification results in terms of the G-mean measure.

Based on the results in Table 4, the GSVM-RU technique improves the G-mean metric from 51% to 75% and indicates a higher confidence level within the process of relating a given testing record with a training record of the minority class (i.e. type-*B*). On the other hand, the overall accuracy and recall for the majority class are decreased, where a larger sample of records is falsely predicted as type-*B*. Regardless of the improvement in the G-mean rate, it is still quite evident that the distinct separation of type-*A* with type-*B* within the training phase of the GSVM-RU technique is still weak. This weakness is evident by the decreasing rate-wise behaviour of the rest of the classification performance metrics. In particular, the overall accuracy rate reached a 75% rate whereas the F-score metric attained 26%. Most intriguing appeared to be the rate obtained for the precision metric that achieved the disappointing value of 16%.

As also evidenced by Fig 5-(a) and Fig 5-(b) the overall confidence in predicting a type A event is slightly decreased and consistent with the classification accuracy percentages dis-

[6] Specificity denotes the true negative rate with an ML classification process that represents the ratio of true negatives(TN) over the sum of true negatives and false positives(FP).

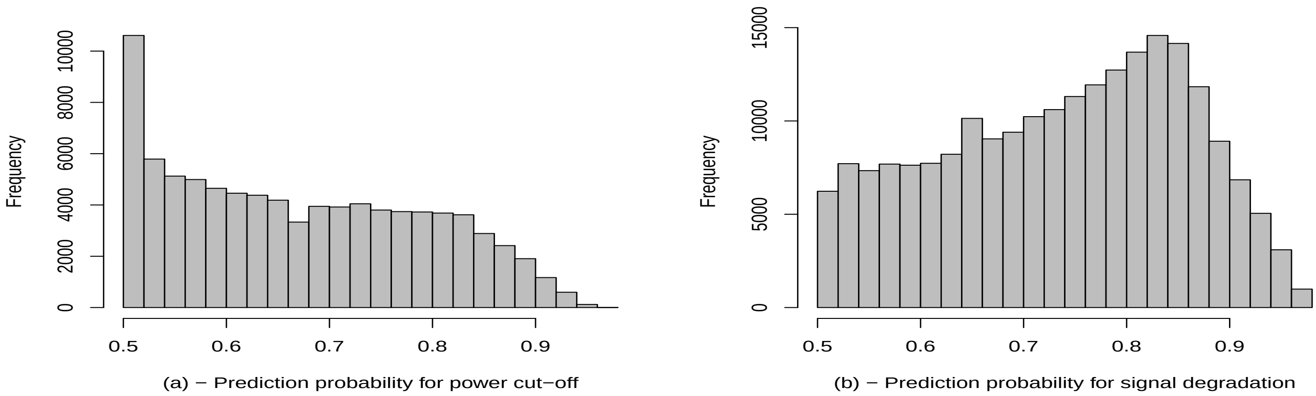

Figure 5: Histogram counts of prediction probabilities when the unlabeled data set is predicted with GSVM-RU (SVM$_2$).

Table 5: Number of correct predictions with the number of iterations of the semi-supervised training algorithm.

Iteration	0	1	2	3	4	5	6	7	8	9	10
TP+TN	15,215	15,217	15,219	15,220	15,220	15,221	15,220	15,220	15,220	15,219	15,221

cussed above. It is also visible that a very high number of samples that are predicted as type B anomalies have a probability just above 0.5 indicating a moderate prediction confidence. We explicitly relate this result with the high number of false positives generated by the GSVM-RU approach. At the same time we argue in fair of the possibility that most of these samples having a probability around 0.5 are actually due to type-A events, or even caused by other, unknown kind of anomalies (e.g. DSLAM monitoring failure or failure of the syslog proto col) that is infeasible for the AD-aware DSLAMs to flag.

Based on the aforementioned results and given the constructive discussion we provide later (section 8) we conclude that the suggested two-class GSVM-RU technique still poses a weakness at confronting the data imbalance problem that exists within our dataset. Through the evaluation of this classifier there was indeed an improvement with respect to G-mean but given the remaining classification metrics this improvement is in practice not useful. Thus, new classification models are required for our dataset.

5.3. *Semi-supervised Two-Class SVM*

In order to address the problems mentioned previously we dedicate this section at presenting our efforts at employing a two-class semi-supervised scheme. Semi-supervised learning in the machine learning community has been widely used in many application domains as reported in [41]. Our semi-supervised SVM algorithm presented here is based on self-training with the main assumption that high confidence predictions should be correct. Therefore in this section, we use both labeled and unlabeled data in order to build robust SVMs that are capable to distinguish type *A* from type *B* events. The main operations of the algorithm are given next.

1. Train a SVM s on a training set of labeled data $\mathcal{L}$.
2. Test s on some test set to get performance metrics.
3. Predict the labels of unknown entries of a set $\mathcal{U}$ using s.
4. Select high confidence predictions, append them in $\mathcal{L}$ with their predicted labels, and remove them from $\mathcal{U}$.
5. Update the model s with the new training set $\mathcal{L}$.
6. Go back to second step.

The number of high confidence predictions selected to update the training set and type can be adjusted according to the generated prediction results. Under the assumption that high confidence predictions are correct, this iterative process enables to refine the model at each iteration. We have applied the described algorithm to the datasets $\mathcal{L}$ and $\mathcal{U}$ using the SVM$_1$ model. At each iteration of the presented semi-supervised algorithm, the top-20 most confident predictions for both types (*A* and *B*) are added to the training set and removed from the unlabelled set. Table 5 illustrates the number of correct predictions (TP+TN) between the iterations 0 and 10. The incorporation of the best predicted unknown labels enables to slightly increase the number of correctly predicted samples in these first iterations. Even if this increase is not significant, this means that the incorporated samples match with the labeled ones justifying the suitability of SVM$_1$ regarding the optimal data description.

The outcomes of this semi-supervised iterative process indicated a slight improvement than those obtained earlier by the fully supervised SVM schemes. Nevertheless, as demonstrated in Table 6 the resulted confusion matrix is not convincing with respect to the problem of data imbalance that resides in our dataset. In comparison with the confusion matrices produced by SVM$_1$ and SVM$_2$ (section 5.2) there is a minimal improvement with respect to the prediction of the class A and B labels. Consequently this aftermath weakens the argument that a semi-supervised scheme would be sufficient for this particular dataset. Therefore the following step in our evaluation was to construct one-class SVM models in order to confront the aspect of data imbalance in our dataset as we show next.

Table 6: Classification performance in terms of confusion matrix obtained with the Semi-supervised SVM.

		True label	
		A	*B*
Predicted label	*A*	14961	723
	B	56	260

6. Construction & Validation of one-class SVM models

6.1. One-class SVM formulation

The supervised one-class SVM algorithm, as proposed by Scholkopf et al. in [43], is an extension of the traditional SVM algorithm. Its main goal is to achieve a decision function capable at returning a class vector y for a given input x based on the distribution of a training dataset. This function is achieved by solving the optimisation problem in Equation 17 using Lagrange multipliers as follows:

$$\begin{aligned} \min_{w,\xi_i,\rho} \ & \tfrac{1}{2}\|w\|^2 + \tfrac{1}{\nu n}\textstyle\sum_{i=1}^{n} \xi_n - \rho \\ & \text{subject to:} \\ (w \cdot \phi(x_i)) & \geq \rho - \xi_i \quad \text{for all } i = 1, \ldots, n \\ \xi_i & \geq 0 \quad \text{for all } i = 1, \ldots, n \end{aligned} \tag{17}$$

In Equation 17, the term ν denotes the solution by setting an upper bound on the fraction of outliers, and a lower bound on the number of support vectors (SVs). By increasing ν it results in a wider soft margin meaning there is a higher probability that the training data will fall outside the normal frontier. The resulting decision function is expressed in Equation 18, where α_i are the Lagrange multipliers.

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) - \rho \tag{18}$$

The function $k(x, x_i)$ represents the kernel function where similarly with earlier we chose to use the RBF kernel (i.e. equation 14, Section 5.1).

6.2. Evaluating one-class SVM classifiers against Decision Trees & Bayesian Networks.

According to table 1 the labels referring to type A events (i.e. signal degradations) are by far the most dominating instances as reported by the AD-aware DSLAMs. Overall, given that the set $\mathcal{S}$ contains a total of 448,544 records, the subset $\mathcal{L}$ holds 135,793 records (i.e. type A events), subset $\mathcal{T}$ denotes type B (i.e. power cuts) events with only 3,300 records whereas subset $\mathcal{U}$ represents the set of unlabelled anomalies as captured by the non-AD aware DSLAMs with a total of 309,451 unlabelled records.

Hence, our main objective was to develop robust one-class models for subsets $\mathcal{L}$ and $\mathcal{T}$. For both $\mathcal{L}$ and $\mathcal{T}$ we've used 70% of their corresponding size in order to construct their classification models and following the suggestions in [9] we scaled our datasets prior the training phase. We have subsequently experimented with several types of kernel mapping functions (e.g. linear, sigmoid, polynomial), SVM-specific tuning parameters (i.e. *nu*, *gamma*) [8] and observed their accuracy performance via a 10-fold cross-validation process. It was revealed that the highest cross-validation percentages for both $\mathcal{L}$ and $\mathcal{T}$ were obtained under a non-linear radial basis function (RBF) kernel with $nu = 0.7$ and $gamma = 0.2$. This is evident by Fig. 6 that illustrates the classification performance with respect to the metrics described earlier (section 3.3). An overall accuracy of more than 95% for both one-class models of type A and B events was achieved. Both classifiers reach a 100% precision rate which in simple terms means that under a testing scenario the labels concluded either as A or B are by a 100% confidence level classified within the correct label. Similarly, the rates for recall, F-score and the 10-fold cross-validation percentages are over 95% and they empower the choice of using these particular SVM models.

A continuing process was to re-validate these models with unlabelled datasets. In particular we used the remaining 30% of sets $\mathcal{L}$ and $\mathcal{T}$ as testing sets in order to measure the performance of our classifiers. The rationale behind this is related with the act of evaluating whether our classifiers would be able to identify only their associated label and exclude any unknown label that did not match any of their training instances. For the purposes of this testing process we referred to the remaining 30% subset of $\mathcal{L}$ as λ whereas the 30% of $\mathcal{T}$ as τ. However, in order to have a correct analogy with respect to the number of records in the testing cases for both classifiers we had to reduce the number of subset λ (i.e. set λ_1 from 40,502 to 981 instances) within the process of testing the type B classifier and also increase the size of τ up to the maximum number of known type B labels (i.e. set τ_1 from 981 to 3,300 instances) while testing the type A classifier.

Table 7 demonstrates the performance of our one-class SVM classifiers over the aforementioned testing sets as well as the results obtained when building training models derived by Decision Trees (DT) and Bayesian Network (BN) as used in [38, 39]. The generated results lean towards the fair conclusion that both one-class classifiers achieve extremely well throughout all the classification performance metrics in contrast with the DT and the BN. In the scenario of identifying signal degradations, the type-A classifier reaches a high overall accuracy of 99.6% , precision of 100%, recall on 99.6% and F-score on a 99.8% rate. Given these results it is undoubtedly evident that this particular classifier has an extremely high probability of correctly labelling a signal degradation event (i.e. recall) under an ensured confidence level that the labelled event definitely bounds within the statistical fingerprint of a signal degradation (i.e. precision). On the other hand, the results obtained whilst testing power cut events (i.e. type-B) through the τ_1 set demonstrate that this particular classifier considers them as outliers (i.e. 0% accuracy), thus it is only in a position to explicitly classify sig nal degradation events. In general, both results attained while testing the λ and τ_1 justify the highly accurate results obtained by the 10-fold cross-validation procedure employed earlier and discussed via fig. 6, hence ensuring the accuracy in our type-A one-class classifier. The type-B classifier reached a 100% of accuracy in all the classification performance metrics and

Table 7: Accuracy and Training Cost Performance for type-A, type-B one-class SVMs, Decision Trees and Bayesian Network training models.

Type-A one-class SVM (Signal Degradations Classifier)					
Test Set	*Accuracy*	*Precision*	*Recall*	*F-score*	*Training Time*
(Sig. Deg.) λ	99.64%	100%	99.68%	99.83%	2.76 seconds
(Power cuts) τ_1	0%	0%	0%	0%	
Type-B one-class SVM (Power Cuts Classifier)					
Test Set	*Accuracy*	*Precision*	*Recall*	*F-score*	*Training Time*
(Power cuts) τ	100%	100%	100%	100%	1.97 seconds
(Sig. Deg.) λ_1	0%	0%	0%	0%	
Decision Tree (two-class)					
Test Set	*Accuracy*	*Precision*	*Recall*	*F-score*	*Training Time*
(Sig. Deg.) $\mathcal{L}$	98%	98%	99%	99%	3.15 seconds
(Power cuts) $\mathcal{T}$	98%	77%	30%	43%	
Bayesian Network (two-class)					
Test Set	*Accuracy*	*Precision*	*Recall*	*F-score*	*Training Time*
(Sig. Deg.) $\mathcal{L}$	98%	98%	99%	99%	3.27 seconds
(Power cuts) $\mathcal{T}$	98%	70%	26%	38%	

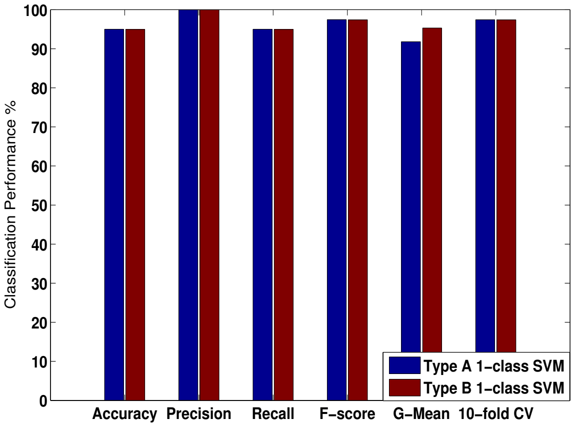

Figure 6: Classification Performance for one-class SVMs for type A (signal degradations) and type B (power cut) events.

similarly with the type-A classifier justified the accuracy rates responding to its training phase (i.e. Fig. 6). At the same time, the unlabelled testing instances of signal degradations were all detected as outliers and did not match with any of the training instances of the type-B classifier. During the composition of the classification models described and presented via Fig. 6 the type-A classifier was trained under a much larger training sample (i.e. 95,325 instances) than that used for the type-B classifier (i.e. 2,356 instances). Despite the fact that one-class SVMs are strongly dependent on the correct selection and tuning of the *gamma* and *nu* parameters, SVMs in their training phase do also depend on the size of the training sample. Given the theoretical formulation of one-class SVMs [8] it was anticipated that a large training sample would naturally invoke a large number of Support Vectors(SVs) on the feature space. Thus, the accuracy rates determined by the distance of these support vectors are directly affected, hence a large number of SVs leads to larger distances between points in the feature space and flawed accuracy outcomes. In our case, the type-A classifier is composed by 81,300 SVs whereas the type-B classifier by just 2,916. Based on the trade-off between the training size and the resulting number of SVs that naturally affect the overall accuracy, we confidently believe that for the case of the type-A classifier we have achieved a robust classification scheme given our validation process.

Despite the high overall accuracy[7] demonstrated by both the DT and the BN models it is quite clear that both lean to classify better the type-A events rather than the type-B events. The inability of both on accurately predicting type-B events (i.e. power cuts) is indicated through the low precision (77% and 70%), recall (30% and 26%) and F-score (43% and 38%) metrics which relate with the fact that their initial training models were based on an extremely large number of known type-A instances. Thus any new testing instance had a much higher probability to be labelled as a type-A rather than a type-B event. Therefore, in contrast with the independent one-class SVMs that confront the data imbalance issue, both the Decision Tree and Bayesian Network classifiers demonstrate a weakness at the data imbalance scheme and lead to flawed prediction outcomes. In parallel to the classification outcomes on the examined training models, we have also assessed the time taken to compute them. As evidenced by Table 7 both one-class SVM classifiers achieved to produce a training model much quicker than the compared BN and DT formulations. Thus, under a realistic close-to real-time deployment, the proposed one-class schemes can respond under a quicker fashion. Overall, these outcomes demonstrate the applicability and robustness of our one-class classifiers at quickly classifying with a high level of accuracy any reported power cut or signal degradation within our DSL infrastructure and ignoring any other types of events in each case. Given the resulted outcomes presented in Table 7 we argue that a synergistic use of the type-A and type-B classifiers

[7] Given the two-class nature of the DT and BN classifiers we only derived the overall weighted accuracy for the overall classifier accuracy.

Table 8: Classification performance of type A one-class SVM under a smaller training sample.

Test Set	Accuracy	Precision	Recall	F-score	10-f CV
$\mathcal{L}_s$	96.15%	100%	95.15%	97.51%	97%

Table 9: Per SyncTrap events summary of classification results for all three different classification approaches.

Classifier	Sig. Deg.	Power Cuts	New Anomalies
Type A & Type B one-class SVMs	154844	154561	46
Decision Tree	340056	4743	0
Bayesian Net	339972	4827	0

overcomes the limitations with respect to data imbalance that previously used techniques such as the DT and BN face. Hence, they can significantly contribute towards the identification and characterization of faults in real DSL deployments where it is highly possible that a large segment of DSLAMs are non-AD enabled as we show next.

7. Inferring unknown events from the non-AD aware DSLAMs

A following step in our evaluation process was to infer the anomalies contained within set $\mathcal{U}$ using the one-class classifiers presented earlier and further compare our findings with the training models of the DT and BN and as they were implemented in [38, 39].

The set $\mathcal{U}$ consists of $309,451$ SyncTrap events that are all reported as unknown events by the non-AD aware DSLAMs. Given the fact that the AD-aware DSLAMs have reported only two types of anomalies (i.e. signal degradations and power cuts) we speculated that all these unknown events would lie within one of the two anomaly types. At the same time, it was anticipated that a number of instances would not lie within the classification margins of neither of the two classifiers, thus they would be detected as outlier events from both.

Prior the pre-processing stage of the set $\mathcal{U}$ we had to re-train a type-A classifier under a training sample size that aligns with the training performed for the type-B classifier. Therefore we extracted a subset of $\mathcal{L}$ (i.e. $\mathcal{L}_s$) that contained an amount of 2356 type A instances and following the same procedure as described in section 6 we re-constructed our new type-A classifier. Similarly with earlier we employed a 10-fold cross validation and achieved high accuracy percentages as depicted by table 8.

Moreover, in order to construct a robust testing phase and keep the analogy between testing and training sets we also had to segment $\mathcal{U}$ into 310 equal subsets u (u_n=$\{u_1$,u_2,u_3,$\cdots$,$u_{310}\}$). Each subset contained 1000 testing records and with respect to the training samples utilized in our classification models we kept an analogy of 70% (training) - 30% (testing) as this is a safe analogy indicated by other pieces of work as in [20]. The outcomes of the classification process for every data segment u were aggregated and summarized with respect to the finalized predicted SyncTrap events for all classifiers as shown in Table 9.

Apart from inferring the known labels, the synergistic employment of the one-class models also derived the existence of unknown events that do not match with any training instance. As shown in table 9 from a total of $309,451$ unknown records in $\mathcal{U}$ we have extracted 46 SyncTrap events that were not classified neither from the type-A nor the type-B classifier. Based on the fact that there are only two known anomalies (i.e. signal degradations and power cuts) on the DSLAM level we argue that the unclassified anomalies are related strictly with hardware fault that could be caused due to wrong configurations on the given DSLAM paths (i.e. clients).

On the other hand and in contrast with the one-class SVM training models, the inferred labelling achieved through the DT and BN models was in a position to provide an extension of the view they had with the known events captured on AD-aware DSLAMs as used within their training phase (i.e. section 6.2). In particular, they have shown the same low proportion of power cuts with 4743 for the DT and 4827 for the BN where the largest amount of unlabelled events were inferred as signal degradations. Given the biased training phase caused by the data imbalance issue where both classifiers had extremely low performance on classifying type-B events (i.e. power cuts) it is quite reasonable to conclude that their prediction capability on unlabelled datasets would also be extremely biased. Hence, this experimentation demonstrated the impact of the data imbalance experienced in the training phase of the DT and BN and its consequent effect on inferring unknown labels.

Overall, the outcomes of this comparison indicated that the simplistic knowledge of only one type of event as it happens with the one-class SVM models, provides the means to construct robust classifiers that can ensure high levels of accuracy when aiming to predict unlabelled datasets. In parallel the extraction of new types of anomalies empowers our argument regarding the applicability of one-class models in order to identify possible novel events. In the context of anomalies on the DSL-level infrastructure, this result sets new promising paths towards investigating new types of anomalies that surely affect the quality of service on the upper-layer IPTV distribution network.

8. Two-class vs. One-class SVMs

Given the comparison with commonly used two-class supervised formulations such as Decision Trees and Bayesian Networks performed previously, this section, aims to further pinpoint some of the benefits derived by using a one-class SVM formulation rather than relying on the traditional two-class SVM

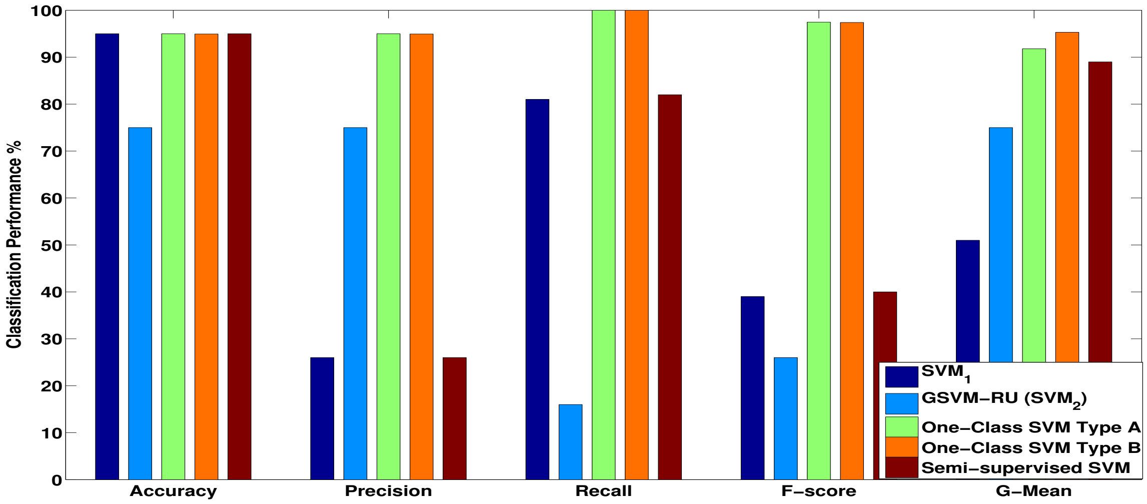

Figure 7: Classification performance metrics obtained within the construction of the SVM-based models

approach. Previous sections have discussed that the classification performance of any ML algorithm heavily depends on the size of records contained within an already known training sample. Undoubtedly, the size relationship between the distinct training labels has proven to be a major constraint in the explicit model construction of two-class approaches. As it was illustrated in sections 5.2 and 5.3 the large data imbalance existing between the two discriminant training classes of type-A and type-B anomalies has negatively affected the training model construction in both two-class supervised and semi-supervised SVM formulations.

Fig. 7 provides a visual comparison with respect to the classification metrics obtained while composing the leaning models by each classifier in this work. Apart from the GSVM-RU (i.e. SVM$_2$) algorithm it is fairly obvious that the rest demonstrate a high overall accuracy rate. By excluding the stable behavior of the one-class SVMs, Fig. 7 illustrates a varying behavior on the other SVM formulations with respect to the values obtained for the remaining classification metrics. In particular, SVM$_1$ and the semi-supervised SVM perform under a really low precision rate whereas they get slightly improved under their recall, F-score and G-mean rates. Still, their performance is clearly lower than that obtained by the type-A and type-B one-class SVMs. In parallel, the GSVM-RU has a much better precision and G-mean rate than SVM$_1$ and the semi-supervised scheme but holds disappointing outcomes on the recall and F-score metrics.

Given the experimental evidence illustrated by Fig. 7 it is undoubtable that the problem assessed within this paper is efficiently resolved under a one-class classification scheme. Both one-class SVM classifiers that are dedicated at only one type of event have produced high accuracy rates. At the same time, the intuition behind their formulations restrict the implications triggered by the data imbalance problem. As presented in this work, the strong data imbalance residing in the datasets has negatively affected the classification performance of the traditional two-class schemes on a supervised and semi-supervised formulation.

9. Conclusions

Undoubtedly, the daily cycle for the management and diagnosis of computer networks relies on the systematic empirical analysis by network operators. A great asset for the effective and efficient diagnosis and management of such networks is considered to be Machine Learning and particularly the classification and clustering techniques derived by this area.

This paper has exhibited an extensive experimental evaluation of Support Vector Machines (SVMs) for the particular task of fault classification in DSL-based networks and provided a comparison with two commonly used ML techniques; the Decision Trees (DT) and Bayesian Networks (BN). Through extensive evaluations using real pre-captured operational DSL Sync-Trap data, this work has proposed the applicability of one-class SVMs as suitable schemes that can sufficiently confront pragmatic scenarios which are experienced in real DSL network deployments.

As explained throughout the paper, in the majority of cases, hardware configurations embodied within the DSLAMs that reside in DSL-based networks, do not provide any meaningful information with respect to the root cause of a failure for a given DSLAM path. By considering this real situation scenario that is repeatedly confronted by network operators, we have proposed one-class SVMs since they can adequately classify and further infer DSL-level events. We show that the one-class SVM formulation overcomes the problem with data imbalance since data imbalance within the training and testing phase forks a negative impact to the classification outcomes of the traditionally used two-class SVM formulations on both a supervised and semi-supervised approach as well as on the commonly used DT and BN schemes. Through this paper it was clearly indicated that the one-class nature of our proposed method is not biased by the aforementioned problem and may produce high scored classification accuracy metrics that reach over a 95% of overall fault classification accuracy on unlabelled events as flagged by improperly configured non-AD DSLAMs.

Our proposed methodology is proven to also identify new types of events. Overall, the experimental outcomes presented in this work broaden the horizons towards the deeper investigation of DSL-level anomalies and grant the properties of one-class SVM classifiers as assets to fault management schemes for networked environments.

References

[1] G., P., Kumar, P. Venkataram, *Artificial Intelligence Approaches to Network Management: recent advances and a survey.*, in Elsevier Computer Communications (COMCOM), vol. 20, 1997

[2] A. Gupta, B.E. Prasad (Eds.), *Principles of Expert Systems*,. IEEE Press, New York, 1988.

[3] A.S. Sethi. Bibliography on network management, ACM Computer Communication Review, 1988

[4] A.A. Covo. T.M. Moruzzi. E.D. Peterson, Al-assisted telecommunication network management, in Proceedings of IEEE GLOBECOM, 1989.

[5] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao,*Towards Automated Performance Diagnosis in Large IPTV Networks*, ACM SIGCOMM 2009

[6] A. Mahimkar, Z. Ge, J. Wang, J. Yates, Y. Zhang, J. Emmons, B. Huntley and M. Stockert, *Rapid Detection of Maintenance Induced Changes in Service Performance*, ACM SIGCOMM CoNEXT 2011

[7] A. K. Marnerides, S. Malinowski, R. Morla, M. R. D. Rodrigues and H. S. Kim, *Towards the Improvement of Diagnostic Metrics: Fault Diagnosis of DSL-Based IPTV Networks using the Renyi Entropy*, IEEE GLOBECOM, 2012

[8] C.-C. Chang and C.-J. Lin, *LIBSVM : a library for support vector machines*, in ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011

[9] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *A practical guide to support vector classification.*, Technical report, Department of Computer Science, National Taiwan University. July, 2003.

[10] Y. Tang, Y-Q. Zhang, N.V. Chawla and S. Krasser, *SVMs modeling for highly imbalanced classification*, IEEE Trans. Sys. Man Cyber. Part B, Feb. 2009.

[11] D. Freedman and P. Diaconis, *On the histogram as a density estimator*, L2 theory. Probability Theory and Related Fields (Heidelberg: Springer Berlin), 1981.

[12] A. K. Marnerides, D. Pezaros, H. Kim and D. Hutchison, *Unsupervised Two-Class and Multi-Class Support Vector Machines for Abnormal Traffic Characterization*, In Proceedings of Passive and Active Measurements (PAM) Conference Student Workshop'09, 2009

[13] A. K. Marnerides, D. Pezaros, H. Kim and D. Hutchison, *Internet Traffic Classification under Energy Time-Frequency Distributions* to appear in IEEE International Conference on Communications, IEEE ICC 2013

[14] A. K. Marnerides, S. Malinowski, R. Morla, M. R. D. Rodrigues and H. S. Kim, *On the Comprehension of DSL Sync Trap Events in IPTV Networks* in the 18th IEEE Symposium on Computers and Communications, IEEE ISCC 2013

[15] A. P. Dempster and N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, in the Journal of the Royal Statistical Society, series B, vol. 39, pp 1-38, 1977

[16] L.E. Baum and T. Petrie and G. Soules and N. Weiss,A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains, Annals of Math. Statistic, vol 41, pp-164-171, 1970

[17] A. Lakhina, M. Crovella, and C. Diot, *Diagnosing network-wide traffic anomalies*, in Proceedings of ACM SIGCOMM 2004, 2004

[18] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, *An Empirical Evaluation of Entropy-based Traffic Anomaly Detection*, in Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, IMC, 2008

[19] Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, L., Internet traffic classification demystified: myths, caveats, and the best practices, ACM CoNEXT, Spain, Madrid, December 9-12, 2008

[20] J. Erman, M. Arlitt, and A. Mahanti, *Traffic classification using clustering algorithms*, in MineNet '06: Proc. 2006 SIGCOMM workshop on Mining network data. New York, NY, USA, ACM Press, 2006

[21] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson, *Identifying and discriminating between web and peer-to-peer traffic in the network core*, in WWW ' 07: Proc. 16th international conference on World Wide Web. Banff, Alberta, Canada, ACM Press, May 2007

[22] V. Paxson, *Bro: A system for detecting network intruders in real-time*, Computer Networks, no. 31(23-24), pp 2435-2463, 1999.

[23] Snort - The de facto standard for intrusion detection/prevention, http://www.snort.org, as of August 14, 2007.

[24] A. Patel, G. McDermott, C. Mulvihill, *Integrating network management and artificial intelligence*, in: B.Meandzija, J.Westcott (Eds.), Integrated Network Management I, North-Holland, Amsterdam, 1989, pp 647-660.

[25] M., I., Steinder, S., S., Adarshpal, *A survey of fault localization techniques in computer networks*, in Elsevier Science of Computer Programming, vol. 53, i 2, November, 2004

[26] M. Klemettinen, H. Mannila, H. Toivonen, *Rule discovery in telecommunication alarm data*, Journal of Network and Systems Management 7 (4) (1999) 395-423.

[27] L. Bernstein, C.M. Yuhas, *Expert systems in network management : the second revolution*, in IEEE J. Select. Areas Commun. (IEEE JSAC) 6 (5) (1988) 784-787.

[28] A. K. Marnerides, D. P. Pezaros and D. Hutchison, *Detection and Mitigation of Abnormal Traffic Behavior in Autonomic Networked Environments*, in ACM SIGCOMM CoNEXT Student Workshop '08, 2008

[29] A. K. Marnerides, D. P. Pezaros and D. Hutchison, *Autonomic Diagnosis of Anomalous Network Traffic*, in 11th IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC), 2010

[30] T.E. Marques, *A symptom driven expert system for isolating and correcting network faults*, in IEEE Commun. Magazine, vol. 26

[31] S. Rabie, A. Rau-Chaplin, T. Shibahara, *DAD: a real-time expert system for monitoring of data packet networks*, in IEEE Network, 2 (5) (1988).

[32] K.-W.E. Lor,*A network diagnostic expert system for Acculink multiplexers based on a general network diagnostic scheme*, in H.G. Hegering, Y. Yemini (Eds.), Integrated Network Management III, North- Holland, Amsterdam, 1993

[33] V., Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[34] H. Drucker, D. Wu, and V. Vapnik, *Support vector machines for spam categorization*, in IEEE Transactions on Neural Networks ,10 5, 1999, pp. 1048-1055.

[35] H.,W., Lilliefors, *On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown*, Journal of the American Statistical Association, vol. 62, issue 318, 1967

[36] L.R. Rabiner, *A tutorial on Hidden Markov models and selected applications in speech recognition*, in Proceedings of the IEEE, Feb. 1989.

[37] Jin, Y., Duffield, N., Gerber, A., Haffner, P., Sen, S., and Zhang, Z-L.,. NEVERMIND, the problem is already fixed: proactively detecting and troubleshooting customer DSL problems. In Proceedings of the 6th ACM CoNEXT, 2010

[38] Zheng, A., X., Lloyd, J., Brewer, E., Failure Diagnosis using decision trees, ACM ICAC 2004, 2004

[39] Deljac, Z., Mostak, R., Stjepanovic, T., "The use of Bayesian networks in recognition of faults causes in the BB networks," MIPRO, 2010 Proceedings of the 33rd International Convention , vol., no., pp.771,775, 24-28 May 2010

[40] M. Aizerman, E. Braverman, and L. Rozonoer, *Theoretical foundations of the potential function method in pattern recognition learning*, in Automation and Remote Control, 1964.

[41] X. Zhu, *Semi-Supervised Learning Literature Survey*, 2006.

[42] P. Langfelder, B. Zhang, and S. Horvath, *Defining clusters from a hierarchical cluster tree*, Bioinformatics, 24:5, March 2008.

[43] B. Scholkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, *Support vector method for novelty detection.*, in NIPS, vol. 12, 1999, pp. 582âĂŞ588.