Statistical Machine Learning and Dissolved Gas Analysis: A Review

Piotr Mirowski, *Member, IEEE,* and Yann LeCun

***Abstract*—Dissolved Gas Analysis (DGA) of the insulation oil of power transformers is an investigative tool to monitor their health and to detect impending failures by recognizing anomalous patterns of DGA concentrations. We handle the failure prediction problem as a simple data mining task on DGA samples, optionally exploiting the transformer's age, nominal power and voltage, and consider two approaches: binary classification and regression of the time-to-failure. We propose a simple logarithmic transform to preprocess DGA data in order to deal with long-tail distributions of concentrations. We also investigate the applicability of semi-supervised learning to exploit the knowledge about the DGA concentration distributions in large unlabeled datasets of transformers. We have reviewed and evaluated 15 standard statistical machine learning algorithms on that task, and reported quantitative results both on a small but published set of power transformers, and on proprietary data from thousands of network transformers of an utility company. Our results confirm that nonlinear decision functions such as neural networks, support vector machines with Gaussian kernels, or local linear regression can theoretically give a slightly better performance than linear classifiers or regressors. Software and part of the data are available at http://www.mirowski.info/pub/dga.**

I. Introduction

DISSOLVED Gas Analysis (DGA) has been used for more than 30 years [1]–[3] for the condition assessment of functioning electrical transformers. DGA measures the concentrations of hydrogen (H_2), methane (CH_4), ethane (C_2H_6), ethylene (C_2H_4), acetylene (C_2H_2), carbon monoxide (CO) and carbon dioxide (CO_2) dissolved in transformer oil. CO and CO_2 are generally associated with the decomposition of cellulosic insulation; usually, small amounts of H_2 and CH_4 would be expected as well. C_2H_6, C_2H_4, C_2H_2 and larger amounts of H_2 and CH_4 are generally associated with the decomposition of oil. All transformers generate some gas during normal operation, but it has become generally accepted that gas generation, above and beyond that observed in normally operating transformers, is due to faults that lead to local overheating or to points of excessive electrical stress that result in discharges or arcing.

A. About the Difficulty of Interpreting DGA Measurements

Despite the fact that DGA has been used for several decades and is a common diagnostic technique for transformers, there are no universally accepted means for interpreting DGA results. IEEE C57-104 [3] and IEC 60599 [4] use threshold values for gas levels. Other methods make use of ratios of gas concentrations [2], [5] and are based on observations that relative gas amounts show some correlation with the type, the location and the severity of the fault. Gas ratio methods allow for some level of problem diagnosis whereas threshold methods focus more on discriminating between normal and abnormal behavior.

The amount of any gas produced in a transformer is expected to be influenced by age, loading and thermal history, the presence of one or more faults, the duration of any faults, and external factors such as voltage surges. The complex relationship between these is, in large part, the reason why there are no universally acceptable means for interpreting DGA results. It is also worth pointing out that the bulk of the work, to date, on DGA has been done on large power transformers. It is not at all clear how gas thresholds, or even gas ratios, would apply to much smaller transformers, such as network transformers, which contain less oil to dilute the gas.

B. Supervised Classification of DGA-based Features

Because of the complex interplay between various factors that lead to gas generation, numerous data-centric machine learning techniques have been introduced for the prediction of transformer failures from DGA data [6]–[17]. These methods rely on DGA samples that are labelled as being taken either on a "healthy" or on a "faulty" (alternatively, failure-prone) transformer. As we will review them in Section II, we will see that it is not obvious, from their description, how each algorithm contributed to a good classification performance, and why should one be specifically chosen over any other. Neither are we aware of comprehensive comparative studies that would benchmark those techniques on a common dataset.

In a departure from previous work, we propose not to add a novel algorithm to the library, but instead to review in Section IV common, well-know statistical learning tools that are readily available to electrical engineers. An extensive computational evaluation of all those techniques is conducted on two different datasets, one (small) public dataset of large-size power transformers (Section V-B), and one large proprietary dataset of thousands of network transformers (Section V-C).

Additionally, the novel contributions of our work lie in the use of a logarithmic transform to handle long-tail distributions of DGA concentrations (Section III-B), in approaching the problem by regressing the time-to-failure, and in considering semi-supervised learning approaches (Section IV-C).

All the techniques previously introduced, as well as those presented in this article, have in common the following steps: 1) the constitution of a dataset of DGA samples (section III-A), and 2) the extraction of mathematical *features* from DGA

P. Mirowski is with the Statistics and Learning Research Department of Alcatel-Lucent Bell Laboratories, Murray Hill, NJ, 07974, USA. Web page: http://www.mirowski.info.

Y. LeCun is with the Courant Institute of Mathematical Sciences, New York University.

data (section III-B), followed by 3) the construction of a classification tool that is trained in supervised way on the labelled features (section IV).

II. Review of Related Work

A. Collection of AI Techniques Employed

We briefly review here previous techniques for transformer failure prediction from DGA. All of them follow the methodology enunciated in Section I-B, consisting in feature extraction from DGA, followed by a classification algorithm.

The majority of them are techniques [6], [7], [9]–[13], [15], [16] built around a feed-forward neural-network classifier, that is also called Multi-Layer Perceptron (MLP) and that we explain in Section IV. Some of these papers introduce further enhancements to the MLP: in particular, neural networks that are run in parallel to an expert system in [10], Wavelet Networks (i.e. neural nets with a wavelet-based feature extraction) in [16], Self-Organizing Polynomial Networks in [9] and Fuzzy Networks in [6], [12], [13], [15].

Several studies [6], [8], [12], [13], [15], [16] resort to *fuzzy logic* [18] when modeling the decision functions. Fuzzy logic enables logical reasoning with continuously-valued predicates (between 0 and 1) instead of binary ones, but this inclusion of uncertainty within the decision function is redundant with the probability theory behind Bayesian reasoning and statistics.

Stochastic optimization techniques such as genetic programming are also used as an additional tool to select features for the classifier in [8], [12], [14], [16], [17].

Finally, Shintemirov et al. [17] conduct a comprehensive comparison between k-nearest neighbors, neural networks and support vector machines (three techniques that we explain in Section IV), each of them combined with genetic programming-based feature selection.

B. Limitations of Previous Methods

1) Insufficient Test Data: Some earlier methods that we reviewed would use a test dataset as small as a few transformers only, on which no statistically significant statistics could be drawn. For instance, [6] evaluate their method on a test set of 3 transformers, and [7] on 10 transformers. Later publications were based on larger test sets of tens or hundreds of DGA samples; however, only [12], [17] employed cross-validation on test data to ensure that their high performance was stable for different splits of train/test data.

2) No Comparative Evaluation with the State-of-the-art: Most of the studies conducted in the aforementioned articles [8]–[10], [12]–[14] compare their algorithms to standard multi-layer neural networks. But only [17] compares itself to two additional techniques, Support Vector Machines (SVM) and k-nearest neighbors, and solely [13] and [15] make numerical comparisons to previous DGA predictive techniques.

3) About the Complexity of Hybrid Techniques: Much of the previous work introduces techniques that are a combination of two different learning algorithms. For instance [17] use Genetic Programming (GP) optimization on top of neural networks or SVM, while [16] use GP in combination with wavelet networks; similarly, [15] build a self-organizing map followed by a neural-fuzzy model. And yet, the DGA datasets generally consist of a few hundred samples of a few (typically 7) noisy gas measurements. Employing complex and highly parametric models on small training sets increases the risk of over-fitting the training data and thereby, of worse "generalization" performance on the out-of-sample test set. This empirical observation has been formalized in terms of minimizing the structural (i.e. model-specific) risk [19], and is often referred to as the *Occam's razor* principle[1]. The additional burden of hybrid learning methods is that one needs to test for the individual contributions of each learning module.

4) Lack of Publicly Available Data: To our knowledge, only [1] provides a dataset of labeled DGA samples and only [15] evaluate their technique on that public dataset. Combined with the complexity of the learning algorithms, the research work documented in other publications becomes more difficult to reproduce.

Capitalizing upon the lessons learned from analyzing the state-of-the-art transformer failure prediction methods, we propose in our paper to evaluate our method on two different datasets (one of them being publicly available), using as large test sets as possible and establishing comparisons among 15 well-known, simple and representative statistical learning algorithms described in section IV.

III. Dissolved Gas Analysis Data

Although dissolved gas analysis measurements of transformer oil provide concentrations of numerous gases, such as nitrogen N_2, we restrict ourselves to key gases suggested in [3], i.e. to hydrogen (H_2), methane (CH_4), ethane (C_2H_6), ethylene (C_2H_4), acetylene (C_2H_2), carbon monoxide (CO) and carbon dioxide (CO_2).

A. (Im)balanced Transformer Datasets

Transformer failures are by definition rare events. Therefore, and similarly to other anomaly detection problems, transformer failure prediction suffers from the lack of data points acquired during (or preceding) failures, relatively to the number of data points acquired in normal operating mode. This data imbalance may impede some statistical learning algorithms: for example, if only 5% of the data points in the dataset correspond to faulty transformers, a trivial classifier could obtain an accuracy of 95% simply by ignoring its inputs and by classifying all data points as normal.

Two strategies are proposed in this paper to balance the faulty and normal data. The first one consists in data resampling for one of the two classes, and may consist in generating new data points for the smaller class: for instance, during experiments on the public Duval dataset, the ranges of DGA measures for normally operating transformers were known, and we randomly generated new data points within those ranges (see Section V-B). The second strategy consists in selecting a subset of existing data, as we did for instance on our second series of experiments (in Section V-C).

[1]The Occam's razor principle could be paraphrased as "all things being considered equal, the simplest explanation is to be preferred".

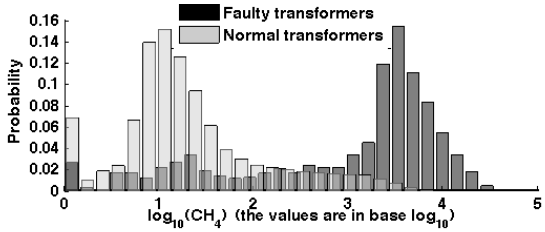

Fig. 1. Histogram of log-concentration of methane CH_4 among samples taken from faulty (red) and normal operating (blue) network transformers (utility data from Section V-C).

B. Pre-processing DGA Data

1) Logarithmic Transform of DGA Concentrations: Dissolved gas concentrations typically present highly skewed distributions, where the majority of the transformers have low concentrations of a few ppm (parts per million), but where faulty transformers can often attain thousands or tens of thousands of ppm [1]–[3]. This fat-tail distribution is at the same time difficult to visualize, and the extreme values can be source of numerical imprecisions and overflows in a statistical learning algorithm.

For this reason, we assert that the most informative feature of DGA data are the order of magnitude of the DGA concentrations, rather than their absolute values. A natural way to account for these changes of magnitude is to rescale DGA data using the logarithmic transform. For ease of interpretation, we used the $\log_{10}$. We assumed that the DGA measuring device might not discriminate between an absence of gas (0 ppm) and a negligible quantity (say 1 ppm), and for this reason, we lower-thresholded all the concentrations at 1 (conveniently, this also avoided us dealing with negative log feature values). We illustrate on Figure 1 how the log-transform can ease the visualization of key gas distributions and even highlight the log-normal distribution of some gases.

2) Relationship to Key Gas Ratios: Conventional methods of DGA interpretation rely on gas ratios [1]–[3]. We notice that log-transforming the DGA concentrations enables to express the ratios as subtractions, e.g.: $\log \frac{[C_2H_2]}{[C_2H_4]} = \log[C_2H_2] - \log[C_2H_4]$. Because most of the parametric algorithms explained in the next section perform at some point linear combinations between their inputs (which are log-transformed concentrations), they may learn to evaluate ratio-like relationships between the raw gas concentrations.

3) Standardizing the DGA Data for Learning Algorithms: In order to keep the numerical operations stable, the values taken by the input features should be close to zero and have a small range of the order of a few units. This requirement stems from the statistical learning algorithms described in the next section, some of whom rely on the assumption that the input data are normally distributed, with a zero mean and unit diagonal covariance matrix. For some other algorithms, such as neural networks, a considerable speed-up in the convergence can be obtained when the mean value of each input variable is close to zero, and the covariance matrix is diagonal and unitary [20]. Therefore, and although we will not de-correlate the DGA measurements, we propose at least to standardize all the features to zero mean and unit variance over the entire dataset. Data standardization simply consists here, for each gas variable X, in subtracting its mean value $E[X]$ over all examples and then dividing the result by the standard deviation $\sqrt{Var[X]}$ of the variable, to obtain $\frac{X-E[X]}{\sqrt{Var[X]}}$. The result of a logarithmic transformation of DGA values, followed by their standardization, is exemplified on Fig. 2, where we plot 167 datapoints (marked as crosses and circles) from a DGA dataset in a two-dimensional space (CH_4 vs. C_2H_4). The ranges of the log-transformed and standardized DGA values on Fig. 2 go from about -2.5 to 2.5 for both gases, with mean values at 0.

C. Additional Features

1) Total Gas: In addition to the concentrations of individual gases, it might be useful to know the total concentration of inflammable carbon-containing gases, that is $[CO + CH_4 + C_2H_2 + C_2H_4 + C_2H_6]$. As with the other concentrations, we suggest to take the $\log_{10}$ of that sum. We immediately see that including this total gas concentration as a feature enables us to express Duval Triangle-like ratios [1], [2], e.g. $\log \%C_2H_2 = \log \frac{[C_2H_2]}{[\texttt{total gas}]} = \log[C_2H_2] - \log[\texttt{total gas}]$.

2) Transformer-specific Features: The age of the transformer (in years), its nominal power (in kVA) and its voltage (in V) are three potential causes for the large variability among transformers' gas production, and could be taken into account for the failure classification task. Because these features are positive and may have a large scale, we also propose to normalize them by taking their $\log_{10}$.

D. Summary: Inputs to the Classifier

At this point, let us note $\mathbf{x_i}$ a vector containing the input features associated to a specific DGA measurement i. These features consist in 7 gas concentrations, optionally enriched by such features as total gas, the transformer's age, its nominal power and voltage. We propose to $\log_{10}$-normalize and to standardize all the features. The next section explains how we find the "label" y_i, and most importantly, how we build a classifier that predicts y_i from $\mathbf{x_i}$.

IV. Methods for Classifying DGA Measurements

This section focuses on our statistical machine learning methodology for transformer failure prediction. We begin by formulating the problem from two possible viewpoints: classification or regression (Section IV-A). Then we recapitulate the most important concepts of predictive learning in Section IV-B before enumerating selected classification and regression algorithms, as well as their semi-supervised version that can exploit unlabeled DGA data points, in Section IV-C. These algorithms are described in more depth in the online Appendix to this paper and are implemented as Matlab code libraries: both are available at http://www.mirowski.info/pub/dga.

A. A Classification or Regression Problem

1) Formulation as a Binary Classification Problem: Although DGA can diagnose multiple reasons for transformer failures [1]–[3] (e.g. high-energy arcing, hot spots above

$400^o\mathtt{C}$, or corona discharges), the primordial task can be expressed as binary classification: "is the transformer at risk of failure?" From a dataset of DGA measures collected on the pool of transformers, one can identify DGA readings recorded shortly before failures, and separate them from historical DGA readings from transformers that kept on operating for an extended period of time. We use the convention that measurement i is labeled $y_i = 0$ in the "faulty" case and $y_i = 1$ in the "normal" case. In the experiments described in the paper, we arbitrarily labeled DGA measurement $\mathbf{x_i}$ as "normal" if it was taken at least 5 years prior to a failure, and "faulty" otherwise.

2) Classifying Measurements Instead of Transformers: As a transformer ages, its risk of failure should increase and the DGA measurements are expected to evolve. Our predictive task therefore shifts from "transformer classification" to "DGA measurement classification", and we associate to each measurement $\mathbf{x_i}$ taken at time t, a label y_i that characterizes the short-term or middle-term risk of failure relative to time t. In the experiments described in the paper, some transformers had more than a single DGA measurement taken across their lifetime (e.g. $\mathbf{x_i}, \mathbf{x_{i+1}}, \ldots$), but we considered the datapoints $(\mathbf{x_i}, y_i), (\mathbf{x_{i+1}}, y_{i+1}), \ldots$ separately.

3) Formulation as a Regression Problem: The second dataset investigated in the paper also contained the time-stamps of DGA measurements, along with information about the time of failure. We used this information to obtain more informative labels $y_i \in [0, 1]$, where $y_i = 0$ would mean "bound to fail", $y_i = 1$ would mean "should not fail in the foreseeable future", and values y_i between those two extremes would quantify the risk of failure. A predictor trained on such dataset could have a real-valued output that would help prioritize the intervention by the utility company[2].

4) Labeled Data for the Regression Problem: We obtained the labels for the regression task in the following way. First, we gathered for each DGA measurement, both the date at which the DGA measurement was taken, and the date at which the corresponding transformer failed, and computed the difference in time, expressed in years. Transformers that had their DGA samples done at the time of or after the failure were given a value of zero, while transformers that did not fail were associated an arbitrary high value. These values corresponded to the Time-To-Failure (TTF) in years. Then, we considered only the DGA samples from transformers that (ultimately) failed, and sorted the TTF in order to compute their empirical Cumulated Distribution Function (CDF). TTFs of zero would correspond to a CDF of zero, while very long TTFs would asymptotically converge to a CDF of one. The CDF can be simply implemented using a sorting algorithm; on a finite set of TTF values, the CDF value itself corresponds to the rank of the sorted value, divided by the number of elements. Our proposed approach to obtain labels for the regression task of the Time-to-failure (TTF) is to employ the values of the CDF as the labels. Under that scheme, all "normal" DGA samples from transformers that did not fail (yet) are simply labeled 1.

[2]Note that many classification algorithms, although trained on binary classes, can provide with probabilities.

B. Commonalities of the Learning Algorithms

1) Supervised Learning of the Predictor: Supervised learning consists in fitting a *predictive* model to a training dataset $(\mathbf{X}, \mathbf{y})$ (which consists here in pairs $\{(\mathbf{x_i}, y_i)\}$ of DGA measurements $\mathbf{x_i}$ and associated risk-of-failure labels y_i). The objective is merely to optimize a "*black-box*" function f such that for each data point $\mathbf{x_i}$, the prediction $\bar{y}_i = f(\mathbf{x_i})$ is as close as possible to the ground truth *target* y_i.

2) Training, Validation and Test Sets: Good statistical machine learning algorithms are capable of extrapolating knowledge and of generalizing it on unseen data points. For this reason, we separate the known data points into a *training* (*in-sample*) set, used to define model f, and a *test* (*out-of-sample*) set, used exclusively to quantify the predictive power of f.

3) Selection of Hyper-parameters by Cross-validation: Most models, including the non-parametric ones, need the specification of a few *hyperparameters* (e.g. the number of nearest neighbors, or the number of hidden units in a neural network); to this effect, a subset of the training data (called the *validation* set) can be set apart during learning, in order to evaluate the quality of fit of the model for various values of the hyperparameters. In our research, we resorted to *cross-validation*, i.e. multiple (here 5-fold) validation on five non-overlapping sets. More specifically, for each choice of hyper-parameters, we performed five cross-validations on five sets that contained each 20% of the available training data, while the remaining 80% would be used to fit the model.

C. Machine Learning Algorithms

1) Classification Techniques: We considered k-Nearest Neighbors (k-NN) [21], C-45 Decision Trees [22], neural networks with one hidden $tanh$ layer [23] and trained by stochastic gradient descent [20], [24], as well as Support Vector Machines [25] with three different types of kernels: linear, polynomial and Gaussian.

Some algorithms strive at defining boundaries that would cut the input space of multivariate DGA measurements into "faulty" or "normal" ones. It is the case of decision trees, neural networks and linear classifiers such as an SVM with linear or polynomial kernel, which can all be likened to the tables of limit concentrations used in [3] to quantify whether a transformer has dissolved gas-in-oil concentrations below safe limits. Instead of pre-determined key gas concentrations or concentration ratios, all these rules are however automatically *learned* from the supplied training data.

The intuition for using k-NN and SVM with Gaussian kernels, can be described as "reasoning by analogy": to assess the risk of a given DGA measurement we compare it to the most similar DGA samples in the database.

2) Regression of the Time-to-Failure: The algorithms that we considered were essentially the regression counterpart to the classification algorithms: Linear Regression and regularized LASSO regression [26] (with linear dependencies between the log-concentrations of gases and the risk of failure), Weighted Kernel Regression [27] (a continuously-valued equivalent of k-NN), Local Linear Regression (LLR) [28], Neural Network Regression and Support Vector Regression (SVR) [29] with linear, polynomial and Gaussian kernels.

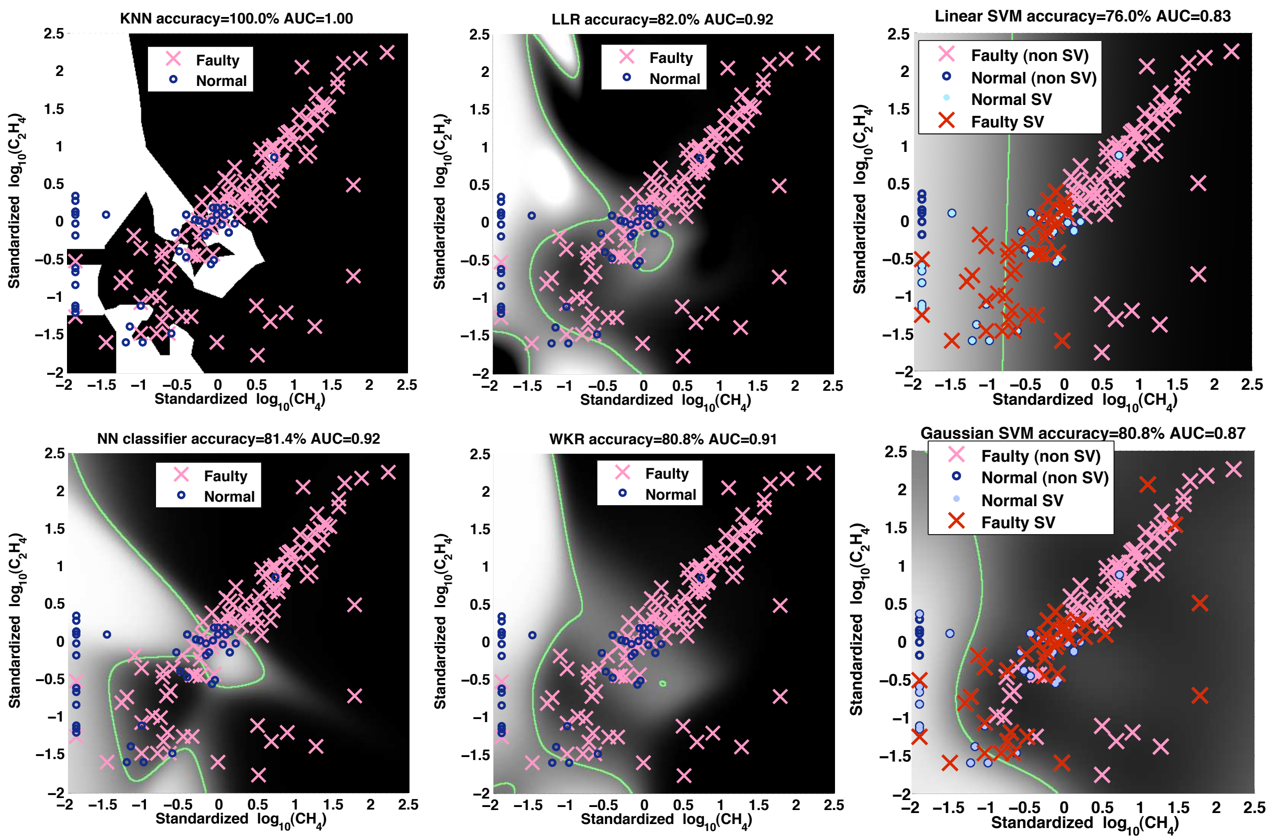

Fig. 2. Comparison of 6 regression or classification techniques on a simplified two-dimensional version of the Duval dataset consisting of log-transformed and standardized values of dissolved gas analysis measures for $\mathtt{CH_4}$ and $\mathtt{C_2H_4}$. There are 167 datapoints: 117 "faulty" DGA measures (marked as red or magenta crosses) and 50 "normal" ones (blue or cyan circles). Because the training datapoints are not easily separable in 2D, the accuracy and Area Under the Curve (see paper) on the training set are generally not 100%. The test data points consist in the entire DGA values space. Output of the 6 decision functions goes from white ($\bar{y} = 1$, meaning "no impending failure predicted") to black ($\bar{y} = 0$, meaning "failure is deemed imminent"); for most classification algorithms, we plot the continuously-valued probability of having $\bar{y} = 1$ instead of the actual binary decision ($\bar{y} = 0$ or $\bar{y} = 1$). The decision boundary (at $\bar{y} = 0.5$) is marked in green. Note that we do not know the actual labels for the test data - this figure provides instead with an intuition of how the classification and regression algorithms operate. k-Nearest Neighbors (KNN, top left) partitions the space in a binary way, according to the Euclidian distances to the training datapoints. Weighted Kernel Regression (WKR, bottom middle) is a smoothed version of KNN, and Local Linear Regression (LLR, top middle) performs linear regression on small neighborhoods, with an overall nonlinear behavior. Neural Networks (bottom left) cut the space into multiple regions. Support Vector Machines (SVM, right) use only a subset of the datapoints (so-called support vectors, in cyan and magenta) to define the decision boundary. Linear kernel SVMs (top right) behave like logistic regression and perform linear classification, while Gaussian kernel SVMs (bottom right) behave like WKR.

3) Semi-Supervised Algorithms: In presence of large amounts of unlabeled data (as was the case for the utility company's dataset explained in the paper), it can be helpful to include them along the labeled data when training the predictor. The intuition behind Semi-Supervised Learning (SSL) is that the learner could get better preparation for the test set "exam" if it knew the distribution of the test data points (aka "questions"). Note that the test set labels (aka "answers") would still not be supplied at training time.

We tested two SSL algorithms that obtained state-of-the-art results on various real-world datasets: Low Dimensional Scaling (LDS) [30], [31] (for classification) and Local Linear Semi-supervised Regression (LLSSR) [32]. Their common point is that they try to place the decision boundary between "faulty" and "normal" DGA samples in regions of the DGA space where there are few (unlabeled, test) DGA samples. This follows the intuition that the decision between a "normal" and "faulty" transformer should not change drastically with small DGA value changes.

4) Illustration on a 2D Toy Dataset: We illustrate on Fig. 2 how a few classification and regression techniques behave on two-dimensional data. We trained six different classifiers or regressors on a two-dimensional, two-gas training set D_{tr} of real DGA data (that we extracted from the seven-gas Duval public dataset, and we plot on Fig. 2 failure prediction results of each algorithm on the entire two-gas DGA subspace. Some algorithms have a linear decision boundary at $\bar{y} = 0.5$, while other ones are non-linear, some smoother than others. For each of the six algorithms, we also report the accuracy on the *training* set D_{tr}. Not all algorithms fit the training data D_{tr} perfectly; as can be seen on these plots, some algorithms obtain very high accuracy on the training set (e.g. 100% for k-NN), whereas their behavior on the entire two-gas DGA space is incorrect; for instance, very low concentrations of both DGA gases, here standardized $\log_{10}(\mathtt{CH_4})$ and $\log_{10}(\mathtt{C_2H_4})$ with values below -1.5, are classified as "faulty" (in black) by k-NN. The explanation is very simple: real DGA data are very noisy and two DGA gases (namely $\mathtt{CH_4}$ and $\mathtt{C_2H_4}$ in this example) are not enough to discriminate well between "faulty" and "normal" transformers. For this reason, we see on Fig. 2 "faulty" datapoints (red crosses) that have very low concentrations of CH_4 and C_2H_4, lower than "normal" datapoints (blue circles): those faulty datapoints may have other gases at much higher concentrations, and we most likely need to consider all seven DGA gases (and perhaps additional information about the transformer) to discriminate well. This figure should also serve as a cautionary tale about the risk of a statistical learning algorithm that overfits the training data but that generalizes poorly on additional test data.

V. Results and Discussion

We compared the classification and regression algorithms on two distinct datasets. One dataset was small but publicly available (see Section V-B), while the second one was large, had time-stamped data, but was proprietary (see Section V-C).

A. Evaluation Metrics

Three different metrics were considered: accuracy, R^2 correlation and Area Under the ROC Curve; each metric had different advantages and limitations.

1) Accuracy (acc): Let us assume that we have a collection of binary (0- or 1-valued) target labels $\mathbf{y} = \{y_i\}$, as well as corresponding predictions $\bar{\mathbf{y}} = \{\bar{y}_i,\}$. When $\bar{\mathbf{y}}$ are not binary but real-valued, we make them binary by thresholding. Then the accuracy of a classifier is simply the percentage of correct predictions over the total number of predictions: 50% means random and 100% is perfect.

2) R^2 correlation: For regression tasks, i.e. when the targets (signal) $\mathbf{y}$ and predictions $\bar{\mathbf{y}}$ are real-valued (e.g. between 0 and 1), the R^2 correlation (equal to $1 - \left(\sum_i(\bar{y}_i - y_i)^2\right) / \left(\sum_i(y_i - E[\mathbf{y}])^2\right)$) quantifies how "aligned" the predictions are with the targets. When the magnitude of the errors $\{y_i - \bar{y}_i\}$ is comparable to the standard deviation of the signal, then $R^2 = 0$. $R^2 = 1$ means perfect predictions. Note that we can still apply this metric when the target is binary.

3) Area Under the ROC Curve: In a binary classifier, the ultimate decision (0 or 1) is often the function of a threshold where one can vary the value of β_0 to get more or fewer "positives" (alarms) $\bar{y}_i = 1$ or, inversely, "negatives" $\bar{y}_i = 0$. Other binary classifiers, such as SVM or logistic regression, can predict the probability $P(y_i = 1|\mathbf{x_i})$ which is then thresholded for the binary choice. Similarly, one can threshold the output of a regressor's prediction $\bar{y}_i$.

The Receiver-Operating Characteristic (ROC) [33] is a graphical plot of the True Positive Rate (TPR) as a function of the False Positive Rate (FPR) as the criterion of the binary classification (the above-mentioned threshold) changes. In the case of DGA-based transformer failure prediction, the true positive rate is the number of data samples predicted as "faulty" and that were indeed faulty, over the total number of faulty transformers, while the false positive rate is the number of false alarms over the total number of "normal" transformers. The Area Under the Curve (AUC) of the ROC can be approximately measured by numerical integration. A random predictor (e.g. an unbiased coin toss) has $TPR \approx FPR$, and we have $AUC = 0.5$, while a perfect predictor first finds all the true positives (e.g. the TPR climbs to 1) before making any false alarms, and thus $AUC = 1$.

Because of the technicalities involved in maintaining a pool of power or network transformers based on periodic DGA samples (namely because a utility company cannot suddenly replace all the risky transformers, but needs to prioritize these replacements based on transformer-specific risks), a real-valued prediction $\bar{y}_i$ is more advantageous than a mere binary classification, as it introduces an order (ranking) of the most risky transformers. The AUC, which evaluates the decision function at different sensitivities (i.e. "thresholds"), is therefore the most appropriate metric.

B. Public "Duval" Dataset of Power Transformers

In a first series of experiments, we compared 15 well-known classification and regression algorithms on a small-size dataset of power transformers [1]. These public data D_{Duval} contain log-transformed DGA values of seven gas concentrations (see Section III) from 117 faulty and 50 functional transformers. Note that because DGA samples in this dataset have no time stamp information, the labels are binary (i.e. $y_i = 0$ "faulty" vs $y_i = 1$ "normal"), even for regression-based predictors. In summary, the input data consisted in $117 + 50 = 167$ pairs $(\mathbf{x}_i, y_i), \forall i \in \{1, 2, \ldots, 167\}$, where each $\mathbf{x}_i \in \mathcal{R}^7$ was a 7-dimensional vector of log-transformed and standardized DGA measurements from 7 gases (see Section III-B).

[1] also provides with ranges of gas concentrations $I \subset \mathcal{R}^7$ for the normal operating mode, which we used to randomly generate 67 additional "normal" data points (beyond the 167 data points from the original dataset) uniformly sampled within that interval. This way, we obtained a new, balanced dataset D^*_{Duval} with $50+67 = 117$ "normal" and 117 "faulty" DGA samples. We evaluating the 15 methods on those new DGA data to investigate the impact of the label imbalance on the prediction performance. For a given dataset D (either D_{Duval} or D^*_{Duval}) and a given algorithm $algo$, we ran the following learning evaluation:

Algorithm 1 $Learn(algo, D)$

Randomly split D (80%,20%) into train/test sets D_{tr}, D_{te}
5-fold cross-validate hyper-parameters θ of $algo$ on D_{tr}
Train algorithm $algo(\theta)$ on $\{(\mathbf{x_i}, y_i)\} \subset D_{tr}$
Test algorithm $algo(\theta)$ on $\{(\mathbf{x_i}, y_i)\} \subset D_{te}$
Obtain predictions $\bar{\mathbf{y}}$ from $\mathbf{X}$ where $\mathbf{X}, \mathbf{y} \subset D_{te}$
Compute Area Under ROC Curve (AUC) of $\bar{\mathbf{y}}$ given $\mathbf{y}$
if classification $algo$ **then** Compute accuracy acc
else Compute correlation R^2

For each algorithm $algo$, we repeated the learning experiment fifty times and computed the mean values of AUC_{algo} as well as acc_{algo} for classification algorithms and R^2_{algo} for regression algorithms. These results are summarized in Table I using the AUC metric and for the original Duval data only (117 "faulty" and 50 "normal" transformers) or after balancing the dataset with 66 additional "normal" DGA data points sampled within I.

TABLE I
PERFORMANCE OF THE CLASSIFICATION AND REGRESSION ALGORITHMS ON THE DUVAL DATASET, MEASURED IN TERMS OF AVERAGE AUC.

Algorithm	Original Dataset D_{Duval}			Balanced Dataset D^*_{Duval}		
	AUC	acc	R^2	AUC	acc	R^2
k-NN		**91%**			**93%**	
C-4.5		85%			88%	
SVM lin.	0.92	85%		0.90	89%	
SVM quad.	0.93	88%		0.93	90%	
SVM gauss.	**0.95**	**90%**		**0.97**	**92%**	
NN log.	0.94	89%		**0.96**	**92%**	
LDS	0.90	88%		**0.96**	**92%**	
Lin. reg.	0.88		0.27	0.88		0.37
LASSO reg.	0.84		0.23	0.58		0.04
NN reg.	0.94		0.48	**0.96**		0.61
SVR quad.	0.94		0.35	0.92		0.35
SVR gauss.	**0.95**		0.44	0.94		0.54
WKR	**0.95**		**0.60**	0.94		**0.70**
LLR	**0.96**		0.55	**0.97**		0.64
LLSSR	0.94		0.43	0.94		0.47

From this extensive evaluation, it appears that the top performing classification algorithms on the Duval dataset are 1) SVM with Gaussian kernels, 2) one hidden-layer neural networks with logistic outputs, 3) k-nearest neighbors (albeit they do not provide probability estimates, which prevents us from evaluating their AUC) and 4) the semi-supervised Low-Dimensional Scaling. These four nonlinear classifiers dominate linear classifiers (here, an SVM with linear kernels) by 3 points of accuracy, suggesting both that the manifold that can separate Duval DGA data is non-linear, and that non-linear methods are more adapted. These results are unsurprising, since Gaussian kernel SVMs and neural networks have proved their applicability and superior performance in many domains.

Similarly, the top regression algorithms in terms of R^2 correlation are the 1) non-parametric Local Linear Regression, 2) single hidden-layer neural networks with linear outputs, 3) SVR with Gaussian kernels, and 4) Weighted Kernel Regression. Again, these four algorithms are non-linear. All of them exploit a notion of local smoothness, but they express a complex decision function in terms of DGA gas concentrations, contrary to linear or Lasso regression.

Finally, we evaluate the impact of an increased fraction f of "normal" data points over the total number of data points. We notice that while the R^2 correlation and the accuracy markedly increase when we balance the data (e.g. from 90% accuracy with unbalanced data to over 96% accuracy with balanced data for Gaussian SVM), with the exception of LASSO regression and SVR with quadratic kernels, the Area Under the Curve does not change as drastically: notably, the AUC of SVM with linear or quadratic kernels, and of most regression algorithms, does not show an upward trend. We can find an obvious explanation for the linear algorithms: the more points are added to the dataset, the less linear the decision boundary, hence the worse the performance of linear classifiers and regressors. We nevertheless advocate for richer (larger) datasets, and conservatively recommend to stick to the data mining rule of thumb of balanced datasets.

C. Large Proprietary Dataset of Network Transformers

1) A Large Dataset of Network Transformers: The second dataset on which we evaluate the algorithms was given by an electrical power company that manages several thousand network transformers.

To constitute our dataset, we gathered time-stamped DGA measures and information about transformers (age, power, voltage, see Section III-C) from two disjoint lists that we call F and N. List F contained 1,796 DGA measures from all transformers that failed or that were under careful monitoring, and list N contained about 30,500 DGA measures from the operating ones. There were about 32,300 DGA measures in total, most conducted within the past 10 years, and some transformers had multiple DGA measures across time.

In the failed transformers list F, we qualified 1,167 DGA measures from transformers that failed because of gas- or pressure-related issues as "positives" and we discarded 629 remaining DGA measures from non-DGA-fault-related corroded transformers. Then, using the difference between the date of the DGA test and the date of failure, we computed a Time-To-Failure (TTF) in years; we further removed 26 transformers that failed more than 5 years since the last DGA test and qualified them as "negatives". Finally, we converted these TTF to numbers between 0 and 1 using the Cumulated Distribution Function (CDF) of the TTF, with values of $y_i = 0$ corresponding to "immediate failure" and values of $y_1 = 1$ corresponding to "failure in 5 or more year".

By definition, transformers in the "normal" transformer list N were not labeled, since they did not fail. We, however, assumed that DGA samples taken taken more than 5 years ago could be considered as "negatives": this represented additional 1,480 data points $\{y_i = 1\}$. The remaining $\sim 29,000$ measurements collected within the last 5 years could not be directly exploited as labeled data.

Like in the public Duval dataset, the input data consisted in pairs $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i \in \mathcal{R}^{11}$ was a 11-dimensional vector of log-transformed and standardized DGA measurements from 7 gases, concatenated with the standardized values of: $\log_{10}[\texttt{total gas}]$, $\log_{10}[\texttt{age in years}]$, $\log_{10}[\texttt{nominal power in kVA}]$ and $\log_{10}[\texttt{voltage in V}]$ (see Sections III-B and III-C), and our dataset D consisted in 2,647 data points, plotted on Figure 3.

2) Comparative Analysis of 12 Predictive Algorithms: We performed the analysis on the proprietary, utility data, similarly to the way we did on the Duval dataset, with the exception that we did not add or remove data points.

We investigated only 12 out of the 15 algorithms previously used, discarding k-Nearest Neighbors and C-45 classification trees (for which one cannot evaluate the AUC) as well as SVR with quadratic kernels (because of computational cost, that was not justified by a mediocre performance on the Duval dataset).

For each algorithm $algo$, we repeated the learning experiment $Learn(algo, D)$ (see Algorithm 1) twenty-five times. We plotted the 25-run average ROC curve on held-out 20% test sets on Figure 4, along with the average AUC curves.

Overall, the classification algorithms performed slightly better than regression algorithm, despite not having access to

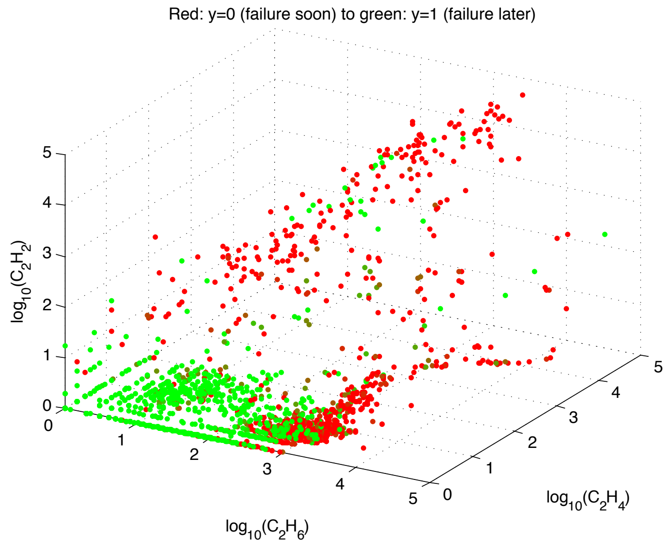

Fig. 3. Three-dimensional plots of DGA samples from the utility dataset, showing log-concentrations of acetylene C_2H2 vs ethylene C_2H_4 and ethane C_2H_6. The color code of the data point labels goes from green/light (failure at a later date, $y = 1$) to red/dark (impending failure, $y = 0$).

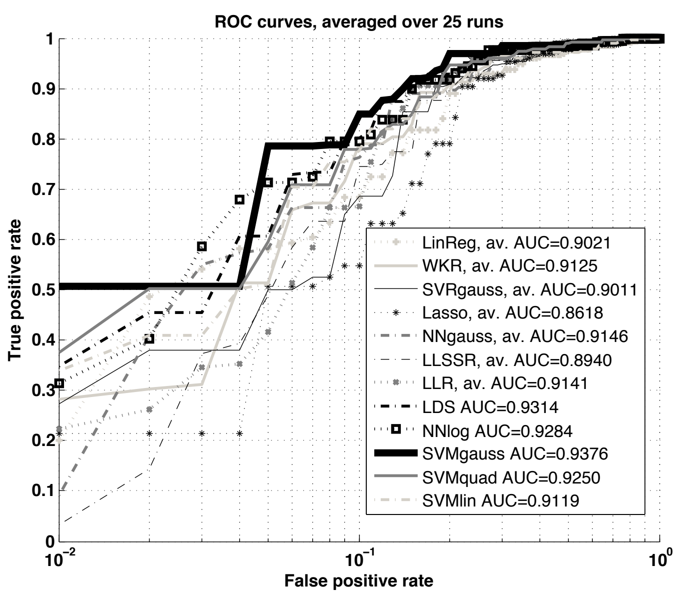

Fig. 4. Comparison of classification and regression techniques on the proprietary, utility dataset. The faulty transformer prediction problem is considered as a retrieval problem, and the Receiver-Operating Characteristic (ROC) is computed for each algorithm, as well as its associated Area Under the Curve (AUC). The learning experiments were repeated 25 times and we show the average ROC curves over all experiments.

subtle information about the Time-To-Failure. The best (classification) algorithms were indeed SVM with Gaussian kernels ($AUC = 0.94$), LDS ($AUC = 0.93$) and neural networks with logistic outputs ($AUC = 0.93$). Linear classifiers or regressors did almost as well as non-linear algorithms.

On one hand, one could deplore the slightly disappointing performance of statistical learning algorithms, as compared to the Duval results, where the best algorithms reached a very high $AUC = 0.97$. On the other hand, this might highlight some crucial differences between the maintenance of small, numerous network transformers and large, scarce power transformers. We conjecture that the data set may have some imprecisions in the labeling, or that we missed some transformer-related discriminative features.

Nevertheless, we demonstrated the applicability of simple, out-of-the-box machine learning algorithms for DGA of network transformers who can achieve promising numerical performance on a large dataset. Indeed, and as visible on Figure 4, at 1% of false alarm rate, between 30% and 50% of faulty DGA samples were detected (using SVM with Gaussian kernels, neural network classifiers or LDS); for the same classifiers and at 10% of false positives, 80% to 85% of faulty DGA samples were detected. This performance still needs to be validated, over an extended period of time, on real-life transformer maintenance.

3) Applicability of Semi-Supervised Algorithms to DGA: In a last, inconclusive, series of experiments, we incorporated knowledge about the distribution of the 29,000 recent DGA measurements. Those were discarded from dataset D because they were not labeled (but they should be mostly taken from "healthy" transformers). We relied on two semi-supervised algorithms (see Section IV-C): Low-Dimensional Scaling (LDS, classification) and Local Linear Semi-Supervised Regression (LLSSR), where unlabeled test data were supplied *at learning time*. The AUC of the semi-supervised algorithms dropped, which can be explained by the fact that the unlabeled test set was probably heavily biased towards "normal" transformers whereas these algorithms are designed for balanced data sets.

VI. Conclusion

We addressed the problem of Dissolved Gas Analysis for the failure prediction of power and network transformers from a statistical machine learning angle. Our predictive tools would take as input log-transformed DGA measurements from a transformer and provide, as an output, the quantification of the risk of an impending failure.

To that effect, we conducted an extensive study on a small but public set of published DGA data samples, and on a very large set of thousands of network transformers belonging to a utility company. We evaluated 15 straightforward algorithms, considering linear and nonlinear algorithms for classification and regression. Nonlinear algorithms performed better than linear ones, hinting at a nonlinear boundary between DGA-samples from "failure-prone" and those from "normal". It was hard to choose between a subset of high-performing algorithms, including Support Vector Machines (SVM) with Gaussian kernels, neural networks, and Local Linear Regression, as their performances were comparable. There seemed to be no specific advantage in trying to regress the Time-To-Failure rather than performing a binary classification; but there was a need to balance the dataset in terms of "faulty" and "normal" DGA samples. Finally, as shown through repeated experiments, a robust classifier such as SVM with Gaussian kernel could achieve an Area Under the ROC Curve of $AUC = 0.97$ on the Duval dataset, and of $AUC = 0.94$ on the utility dataset, making this DGA-based tool applicable to prioritizing repairs and replacements of network transformers.

We have made our Matlab code and part of the dataset available at http://www.mirowski.info/pub/dga in order to ensure reproducibility and to help advance the field.

Acknowledgment

The authors would like to express their gratitude to Profs. W. Zurawsky and D. Czarkowski for their valuable input and help in the elaboration of this manuscript. They would also like to thank the utility company who provided them with DGA data, as well as three anonymous reviewers for their feedback.

References

[1] M. Duval and A. dePablo, "Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases," *IEEE Electrical Insulation Magazine*, vol. 17, pp. 31–41, 2001.

[2] M. Duval, "Dissolved gas analysis: It can save your transformer," *IEEE Electrical Insulation Magazine*, vol. 5, pp. 22–27, 1989.

[3] *IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers*, IEEE Std. C57.104-2008, 2009.

[4] *Mineral Oil-Impregnated Equipment in Service Guide to the Interpretation of Dissolved and Free Gases Analysis*, IEC Std. Publication 60 599, 1999.

[5] R. R. Rogers, "IEEE and IEC codes to interpret incipient faults in transformers, using gas in oil analysis," *IEEE Transactions on Electrical Insulation*, vol. EI-13, pp. 349–354, 1978.

[6] J. J. Dukarm, "Tranformer oil diagnosis using fuzzy logic and neural networks," in *CCECE/CCGEI*, 1993, pp. 329–332.

[7] Y. Zhang, X. Ding, Y. Liu, and P. Griffin, "An artificial neural network approach to transformer fault diagnosis," *IEEE Transactions on Power Delivery*, vol. 11, pp. 1836–1841, 1996.

[8] Y.-C. Huang, H.-T. Yang, and C.-L. Huang, "Developing a new transformer fault diagnosis system through evolutionary fuzzy logic," *IEEE Transactions on Power Delivery*, vol. 12, pp. 761–767, 1997.

[9] H.-T. Yang and Y.-C. Huang, "Intelligent decision support for diagnosis of incipient transformer faults using self-organizing polynomial networks," *IEEE Transactions on Power Delivery*, vol. 13, pp. 946–952, 1998.

[10] Z. Wang, Y. Liu, and P. J. Griffin, "A combined ann and expert system tool for transformer fault diagnosis," *IEEE Transactions on Power Delivery*, vol. 13, pp. 1224–1229, 1998.

[11] J. Guardado, J. Naredo, P. Moreno, and C. Fuerte, "A comparative study of neural network efficiency in power transformers diagnosis using dissolved gas analysis," *IEEE Transactions on Power Delivery*, vol. 16, pp. 643–647, 2001.

[12] Y.-C. Huang, "Evolving neural nets for fault diagnosis of power transformers," *IEEE Transactions on Power Delivery*, vol. 18, pp. 843–848, 2003.

[13] V. Miranda and A. R. G. Castro, "Improving the IEC table for transformer failure diagnosis with knowledge extraction from neural networks," *IEEE Transactions on Power Delivery*, vol. 20, pp. 2509–2516, 2005.

[14] X. Hao and S. Cai-Xin, "Artificial immune network classification algorithm for fault diagnosis of power transformer," *IEEE Transactions on Power Delivery*, vol. 22, pp. 930–935, 2007.

[15] R. Naresh, V. Sharma, and M. Vashisth, "An integrated neural fuzzy approach for fault diagnosis of transformers," *IEEE Transactions on Power Delivery*, vol. 23, pp. 2017–2024, 2008.

[16] W. Chen, C. Pan, Y. Yun, and Y. Liu, "Wavelet networks in power transformers diagnosis using dissolved gas analysis," *IEEE Transactions on Power Delivery*, vol. 24, pp. 187–194, 2009.

[17] A. Shintemirov, W. Tang, and Q. Wu, "Power transformer fault classification based on dissolved gas analysis by implementing bootstrap and genetic programming," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 39, 2009.

[18] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

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

[20] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient backprop," in *Lecture Notes in Computer Science*. Berlin/Heidelberg: Springer, 1998.

[21] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, 1967.

[22] J. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufman, 1993.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[24] L. Bottou, "Stochastic learning," in *Advanced Lectures on Machine Learning*, O. B. et al. (Eds.), Ed. Berlin Heidelberg: Springer-Verlag, 2004, pp. 146–168.

[25] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, 1995.

[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistics Society*, vol. 58, pp. 267–288, 2006.

[27] E. Nadaraya, "On estimating regression," *Theory of Probability and Its Applications*, vol. 9, pp. 141–142, 1964.

[28] C. Stone, "Consistent nonparametric regression," *Annals of Statistics*, vol. 5, pp. 595–620, 1977.

[29] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004.

[30] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the Conference on Artificial Intelligence and Statistics AISTATS'05*, 2005.

[31] A. N. Erkan and Y. Altun, "Semi-supervised learning via generalized maximum entropy," in *Proceedings of the Conference on Artificial Intelligence and Statistics AISTATS'10*, 2010.

[32] M. R. Rwebangira and J. Lafferty, "Local linear semi-supervised regression," School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213, Tech. Rep. CMU-CS-09-106, Feb. 2009.

[33] D. Green and J. Swets, *Signal Detection Theory and Psychophysics*. New York, NY: John Wiley and Sons, 1966.

Piotr Mirowski Piotr Mirowski received a Diplôme d'Ingénieur (2002) from Ecole Nationale Supérieure ENSEEIHT in Toulouse, France, then an M.Sc. (2007) and Ph.D. (2011) from the Courant Institute at New York University (NYU), all in Computer Science. His machine learning thesis was on time series modeling and his advisor was Prof. Yann LeCun. P.M. joined the Statistics and Learning Research group at Bell Labs (Alcatel-Lucent) in 2011; he also worked at Schlumberger Research in Cambridge, England and in Ridgefield, CT (2002-2005), and interned at the NYU Medical Center, Google, Standard & Poor's and AT&T Labs during his Ph.D. He authored 5 patents and patent publications as well as papers on the applications of machine learning to geology, epileptic seizure prediction, statistical language modeling, robotics and indoor localization.

Yann LeCun Yann LeCun was born near Paris, France, in 1960. He received a Diplôme d'Ingénieur from the Ecole Superieure d'Ingénieur en Electrotechnique et Electronique (ESIEE), Paris in 1983, and a PhD in Computer Science from Université Pierre et Marie Curie in 1987 during which he proposed an early form of the back-propagation learning algorithm for neural networks. He was a postdoctoral research associate in Geoffrey Hinton's lab at the University of Toronto.

In 1988, he joined the Adaptive Systems Research Department at AT&T Bell Laboratories in Holmdel, New Jersey, USA, where he developed a number of new machine learning methods, such as a biologically inspired model of image recognition called Convolutional Neural Networks, the "Optimal Brain Damage" regularization methods, and the Graph Transformer Networks method (similar to conditional random field), which he applied to handwriting recognition and OCR. The bank check recognition system that he helped develop was widely deployed by NCR and other companies, reading over 10% of all the checks in the US in the late 1990s and early 2000s.

In 1996, he joined AT&T Labs-Research as head of the Image Processing Research Department, which was part of Lawrence Rabiner's Speech