Integrated Optimization of Long-Range Underwater Signal Detection, Feature Extraction, and Classification for Nuclear Treaty Monitoring

Matthias Tuma, Valdemar Rørbech, Mark K. Prior, and Christian Igel, *Senior Member, IEEE*

***Abstract*—We designed and jointly optimized an integrated signal processing chain for detection and classification of long-range passive-acoustic underwater signals recorded by the global geophysical monitoring network of the Comprehensive Nuclear-Test-Ban Treaty Organization. Starting at the level of raw waveform data, a processing chain of signal detection, feature extraction, and signal classification was designed and jointly optimized to the task. Relevant waveform segments were in a first step identified by a generic, flexibly parameterized detection algorithm on a long- to short-term averages' ratio of the spectral energy. For representation, general-purpose sound processing features, with an added focus on spectral and cepstral features, were extracted from the detected segments. As classifiers, support vector machines with different kernel functions were employed alongside other baseline learning algorithms. The free parameters of the overall toolchain (i.e., trigger algorithm parameters and classifier hyperparameters) were jointly optimized in a cross-validation setting, either according to the cross-validation classification error or the cross-validation area under the receiver operating characteristic curve. Experiments demonstrate that our method outperforms machine learning algorithms task-tailored to a previous, human-expert-designed preprocessing chain. The presented approach can be adapted to a wide range of problems that can benefit from jointly optimizing parameters of preprocessing and classification algorithm.**

***Index Terms*—Acoustic signal detection, Adaptive signal processing, Classification algorithms, Pattern recognition, Underwater acoustics.**

I. Introduction

We devised and jointly optimized a combined processing chain for detection, feature extraction, and classification of hydroacoustic signals recorded by the global geophysical monitoring network of the Preparatory Commission for the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO). The CTBTO's hydroacoustic subnetwork uses in-ocean acoustic sensors, placed at depths of the local SOFAR channel axis [1], to monitor Earth's oceans for evidence of underwater nuclear explosions. Two previous studies on machine learning approaches for classification of CTBTO hydroacoustic signals operate on features devised and extracted by the CTBTO [2], [3]. Both studies task-tailor machine learning methods to accommodate for missing features in the predefined data set (in which less than 5% of samples have a complete feature vector). In contrast, the present study starts at the level of the raw waveform data from which the previous data set originated. We construct a generic architecture for signal detection, feature extraction, and signal classification and adapt the entire processing chain's free parameters in a joint cross-validation setting. In this sense, the present study aims to explore the potential of an alternatively structured and optimized processing chain for CTBT hydroacoustic signal classification. At the same time, it also represents a generic setup for (geophysical) waveform processing and classification in that it optimizes the detection algorithm parameters jointly with the classifier hyperparameters. Despite its structural clarity, such an approach has, to our best knowledge, not been used in trigger algorithm optimization for related geophysical applications, which often utilize well-established parameter values, heuristics, or expert manual tuning [4], [5]. In an organizationally similar setting, a joint optimization of representation and support vector machine (SVM) classifier is presented in [6].

In Section II, we give an overview of the CTBTO's geophysical verification system, its data processing aspects, and implications for later design decisions. Section III describes our integrated and jointly optimized processing scheme for CTBTO hydroacoustic data. It discusses the individual processing chain components, experimental setup, and data attributes used for experiments. Experimental results are presented and discussed in Section IV. An overall discussion and directions for future work conclude this paper.

II. The International Monitoring System

The Comprehensive Nuclear-Test-Ban Treaty (CTBT) for the first time in history foresees a universal legal ban on all nuclear explosions in any setting for any purpose. The key technical instrument for CTBT verification is the CTBTO's globe-spanning International Monitoring System (IMS) [7], which

Manuscript received March 29, 2015; revised July 16, 2015 and November 1, 2015; accepted December 16, 2015. Date of publication March 28, 2016; date of current version April 27, 2016.

M. Tuma is with the Institute for Neural Computation (INI), Ruhr-University Bochum, 44801 Bochum, Germany (e-mail: matthias.tuma@rub.de).

V. Rørbech was with the Department of Computer Science, University of Copenhagen, 2100 Copenhagen, Denmark. He is now with NetCompany A/S, 2100 Copenhagen, Denmark.

M. K. Prior was with the Preparatory Commission for the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO), 1400 Vienna, Austria. He is now with TNO, 2595 The Hague, The Netherlands.

C. Igel is with the Department of Computer Science (DIKU), University of Copenhagen, 2100 Copenhagen, Denmark.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2522972

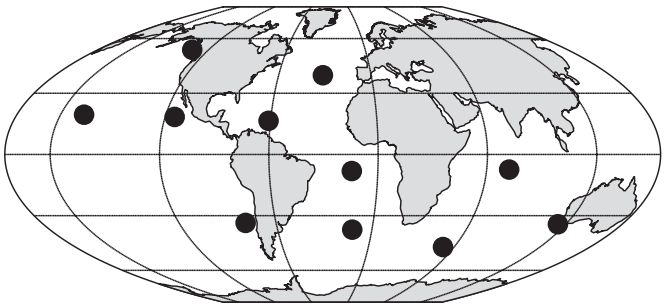

Fig. 1. Locations of hydroacoustic stations defined by the treaty. Sensor stations are either installed on steep-sloped shores of mid-ocean islands or submerged offshore of mid-ocean islands at depths of around 1 km.

collects geophysical data in order to monitor Earth for nuclear explosions (atmospheric, underground, and underwater).

The official treaty text for the CTBT [8] specifies the types, number, and location of IMS sensor stations while additionally imposing some constraints on data recording and/or processing. For example, different high-level feature groups permissible in data processing are given in the treaty. Less formally established, but of equal relevance, are standing operating requirements on, for example, algorithm clarity and precise reproducibility of algorithmic outcomes—which can thus concern models constructed by machine learning algorithms. Within such technical and formal constraints, the verification system is to provide maximum coverage of the Earth and capability for detecting evidence of any nuclear explosion having occurred.

A. Sensor Technology

The IMS foresees a network of 321 geophysical monitoring stations, with the majority of these being seismic stations (170), followed by radionuclide detectors for particulates and noble gases (80), infrasound arrays (60), and hydroacoustic sensors (11). The three regimes of seismic, hydroacoustic, and infrasound monitoring constitute the so-called "waveform" technologies, measuring energy propagated through the solid earth, oceans, and atmosphere, respectively. As of January 2016, the IMS consists of 270 certified sensors (out of the 321 planned), and 10 (out of the 11 planned) hydroacoustic sensors are in certified operations. In full operation, around 15 GB of incoming data is expected daily, mostly transmitted in real time through a global VSAT satellite network. This amount of incoming data makes reliable automatic processing mandatory and is one of the many factors fostering the exploration of adaptive signal processing and machine learning solutions for IMS data processing.

Fig. 1 shows the locations of the 11 hydroacoustic stations as defined by the treaty. Every (either on- or offshore) station consists of several individual sensors, facing different sides of the island that hosts its communication infrastructure. Raw sensor data are then transmitted to the CTBTO's International Data Centre (IDC) in Vienna, Austria, for processing and analysis. The relevant frequency range for passive long-range hydroacoustics is between 1 and 100 Hz, with IMS hydrophones sampling at 250 Hz.

B. Processing of IMS Data

In preparation for entry into force of the treaty and during build-up of the IMS, the CTBTO is tasked with continuing analysis of incoming data already today. The automated IDC algorithms for the three waveform technologies use similar processing flows: in a first phase, data are processed at the level of individual IMS stations (each of which may or may not be an array of individual sensors). This begins with continuous monitoring of the ambient noise stream for signals. When a signal is detected (i.e., when the detection algorithm "triggered"), representative features are extracted from it. These are next used to categorize the signal, where the list of possible classes depends on the sensing technology.

In a second phase called *global association*, detected and categorized signals are cross-processed between sensor stations (and also sensor technologies). All individual detections within certain time windows become associated with postulated hypothesis source events physically able, and most likely, to have caused them. The current global association system in an intricate way combines expert experience, beam forming, physics of wave propagation, and iterative least-squares inversion [9]. The IDC processing routines autonomously reach a first set of hypotheses of what events caused which observations at what station. All claimed events (ca. 100–200 per day) and associated detections are then reviewed by human analysts, who refine or correct about half of them, as well as create new ones (see also [10]).

Within that overall processing scheme, this paper is dedicated to one subtask of automatic hydroacoustic station-level processing, namely, the classification of hydroacoustic signals from single-sensor stations into two categories (nonexplosive or explosive-like). The IDC's current categorization module for hydroacoustic signals is a rule-based expert system. Two previous related studies explore machine-learning-based classifiers for CTBTO hydroacoustic signal classification [2], [3]. Both operate on feature sets computed by the IDC and deploy highly task-specific variants of machine learning algorithms in order to account for missing feature information in the data set. In contrast, the current study designed and jointly optimized a processing chain starting from the level of raw waveform data in a more generic and transferable setting. It thus aims to explore the potential of optimizing to the task at hand a generic abstraction of the IDC's current processing chain which does not rely on machine learning approaches for missing data problems. It seeks to stay as structurally close as possible to current IDC processing in order to facilitate both comparability and further side-by-side evaluation.

The current work presented here has a number of sibling projects, likewise established over the last five years (see [11] and [12] for a broad overview), which also apply classification and other machine learning algorithms for a broad range of well-defined tasks in IMS data processing. A nonexhaustive list of recent studies includes the training of dynamic Bayesian networks on wavelet representations of IMS seismic waveforms [13], the identification of seismic aftershock sequences by using a template-based correlation detector in the time domain [14], or Bayesian inference for detection and localization of seismic events framed as an open universe probability model [10].

Another emerging field of research concerns CTBT on-site inspections, which are the final verification measure taking place at the coordinates of a suspected nuclear test site. Here, Sick *et al.* use a clustering approach for processing extremely low-magnitude seismic data obtained during on-site inspections [15]. As such, the present contribution is one among a progressing series of research on adaptive signal processing for sensor data from the CTBTO's verification network.

III. Optimized Processing Chain for Hydroacoustic Signal Classification

Hydroacoustic processing of IMS data at single-sensor level includes a signal classification step which serves to flag signals with explosive-like signature. Explosive-like events in this sense are not limited to large uninhibited underwater nuclear explosions but include eruptions of underwater volcanoes, dynamite fishing, seismic airgun surveys, military exercises, and accidental explosions, among others. IDC hydroacoustic processing further identifies three nonexplosive categories: earthquakes (waterborne propagation path), earthquakes (crustal path), and noise (all other sources). For the purpose of this study, the latter three categories are regarded as one joint nonexplosive, or noise, class. Examples of such nonexplosive detections are thus those generated by earth- or seaquakes, iceberg calving, shipping activity, or marine mammals. Noise in this sense hence refers to the broader class of nonexplosive signals, even when not noise in the stricter sense of signal processing terminology.

A. Processing Chain Layout

Three components can be identified as structurally relevant to the task of classifying IMS hydroacoustic signals: first, a detection or trigger algorithm, which identifies all incoming signals (explosive-like and noise-type) among the backdrop of ambient noise; second, a feature extraction routine, which transforms the detected signal into a condensed representation better suited for classification; and third, the discriminative stage, in which a classifier, operating on the extracted features, casts vote on the source type of origin (explosive or nonexplosive). Note that the boundary between trigger and classifier cannot always be well defined. To provide a processing flow as compatible with current IDC processing as possible, the present study structurally follows the IDC's setup, which implements the three-stage framework outlined previously. The IDC's discrimination algorithms for hydroacoustic signals are mildly parameterized rule-based expert systems, with its parameters partially inferred from existing data.

In this paper, the parameters of our main processing stages were jointly optimized according to one mutual objective commonly used for hyperparameter optimization in machine learning. Cross-validation was used [16], and depending on the experiment, either the classification error or the area under the receiver operating characteristic (ROC) curve [17] on hold-out data were considered as the optimization criteria. While cross-validation is a standard procedure for SVM model selection, it is, to our best knowledge, not common to jointly optimize trigger algorithm (or comparable preprocessing) parameters according to the same objective. However, our very general layout imposes few structural limitations on the processing chain and may serve as blueprint for other applications requiring optimized signal detection and classification.

B. Main Components—Overview

For detection, we adhere to a well-established and often-used class of triggers based on a threshold on the ratio of short- to long-term energy averages (STA/LTA triggers [4], [5], [18]—where STA and LTA refer to short- and long-term averages, respectively, and the term STA/LTA refers to the ratio of a short- to long-term average). These are implemented in the current IDC system, motivating us to use a generalized variant thereof. Alternatives would have been, for example, the optimization of general time-frequency representations jointly with the trigger and classifier operating on them (see, e.g., [6], [19], and [20]). For representation, we use general-purpose sound processing features with an added emphasis on spectral and cepstral features while still reasonably overlapping with the IDC's current feature set [21].

For signal discrimination, we employ SVMs [22], [23] with linear, radial basis function (RBF), and automatic relevance detection (ARD) kernels (see the following). We also evaluate nearest neighbor (NN) and linear discriminant analysis (LDA) classification as baseline comparison approaches. The search for trigger parameters and classifier hyperparameters is carried out jointly according to a low cross-validation classification error. Alternatively, we also explore the area under the ROC curve as an optimization criterion in the joint search for trigger and classifier parameters.

C. Data Basis and Preparation

A small and well-reviewed reference data set was specifically assembled by the IDC to support research on machine learning for CTBTO hydroacoustic signal classification (see also [18], [24], and [25]). That reference data set consists of 778 data samples, all of which triggered a detection in the original IMS processing system. All detections were again screened for the reference data set's construction by a human expert, even though they had previously been routine-processed by the IDC's system (which already ensures correct labeling in its final human analyst stage). This additional step of quality control reflects the relevance of having a reliably and consistently labeled data set as basis for studies which might inform decisions on future developments of the automatic processing system. For example, rescreening ensures that borderline cases between classes are decided in a consistent manner. Human labeling of detections (both as part of the IDC routine processing and the present additional screening) is predominantly informed by the analyst's visual inspection of a spectral representation of the waveform segment in question. For example, harmonics from gas bubble oscillations following underwater explosions can have a clear visual correspondence as scalloping effects in a signal's spectrum [18]. The analyst may, in addition, consult network information on other stations' arrivals across

the network, as well as on system-postulated source events. In the case of explosive events, analysts may also make use of or investigate independent supporting evidence. For example, industrial or other accidents may be well documented externally, both spatially and temporally. Likewise, external information on airgun surveys or similar activities may be available.

In the compiled reference data set, 307 examples (or 40%) come from explosion-like sources, and 471 examples (or 60%) are of nonexplosive origins (and, as stated previously, all of these triggered a detection in the IDC's original processing system). This close-to-even ratio between the two classes does not reflect the true proportion at which both classes are encountered in routine IDC processing operations, but it is the result of undersampling the majority (nonexplosive) class during IDC data set construction (see Section III-F for a discussion). Those samples labeled as nonexplosive may not contain any signal at all (i.e., false positives by the original detection algorithm) or represent signals from nonexplosive origins. Each hydroacoustic sensor recording in this reference data set was extracted from the continuous data stream in such a way that a constant 100 s of original ambient noise was preserved before the detection, and varying lengths after it.

D. Trigger Algorithm

Triggers in the STA/LTA class are, despite their structural simplicity, both a *de facto* standard in geophysical signal processing [4] and currently used in several IDC processing components. In consequence, we define a generic, or canonically parameterized variant of STA/LTA trigger and then ask the question under which parameter values it best lends itself to an ensuing classification of the signal snippets extracted by it. Instead of postulating a surrogate or intermediate objective, we approach the question directly by including the trigger parameter optimization in the optimization of the classifier hyperparameter values, namely, according to a low cross-validation classification error on the training set.

In its most basic form, an STA/LTA detector at each time step t computes the normalized energy STA/LTA ratio r_x of a discrete-time signal $x(t)$ as

$$r_x(t) = \frac{N_\ell}{N_s} \frac{\sum_{\tau=t-N_s+1}^{t} x(\tau)^2}{\sum_{\tau=t-N_\ell+1}^{t} x(\tau)^2} \tag{1}$$

where N_s and N_ℓ are the short- and long-term window lengths, respectively. Given a predefined detection threshold r_S, a detection is declared whenever and as long as $r_x > r_S$. This basic trigger has the three parameters N_s, N_ℓ, and r_S. A range of specializations exists [4]: e.g., x may be mapped to $\hat{x}$ via some preprocessing filter before calculating $r_{\hat{x}}(t)$ [26]. Other enhancements mitigate the effect of "shadow zones," which cause detections to terminate too early if signals last longer than N_ℓ [4]. One such strategy uses a second threshold $r_R < r_S$, below which $r_{\hat{x}}$ must fall before a detection is terminated. The LTA window can also be delayed with respect to the STA window by an offset, or delay, N_d [5]. Furthermore, detections may always be prolonged by a fixed amount N_z, the post-event time. Likewise, a pre-event time of length N_a can be used to extract representative noise before the arrival. Incorporating these additions yields

$$r_{\hat{x}}(t) = \frac{N_\ell}{N_s} \frac{\sum_{\tau=t-N_s+1}^{t} \hat{x}(\tau)^2}{\sum_{i=t-N_d-N_\ell+1}^{t-N_d} \hat{x}(\tau)^2}. \tag{2}$$

In our setup, we let N_a and N_z share the same value. This generalized STA/LTA trigger thus has six real values and one mapping function $x \mapsto \hat{x}$ as degrees of parameterization. Table I lists each parameter with the values considered in this study (see the following).

TABLE I
PARAMETERS OF THE TRIGGER ALGORITHM USED AND VALUES CONSIDERED

Parameter	Symbol	Grid values
STA window length	N_s	5, 10, 15, 20, 25 s
LTA window length	N_ℓ	40, 50, 60, 70, 80 s
LTA delay	N_d	0, 20, 35, 50 s
Trigger threshold	r_S	1.1, 1.2, 1.3, 1.4, 1.5
Release threshold (relative to r_S)	r_R	0.0, -0.1, -0.2, -0.3, -0.4
Pre- and post-event time	N_a, N_z	0, 5, 10 s

In practice, STA/LTA parameters are often set by hand. For example, fixed values have been passed down in the literature, with additional heuristics like allowing the window lengths to adapt to the zero-crossing rate [5]. Also, manual tuning can be conducted on example signals until the detections approach those of a human analyst [4] (one might imagine automating the latter in a least-squares setting). We here combine trigger parameter optimization with the joint final objective of minimizing the expected generalization error for signal classification. The coarse parameter ranges of Table I were first determined by structural and computational considerations, by variations on common values, and by visual inspection. We likewise conducted preliminary experiments with different preprocessing filters $x \mapsto \hat{x}$ [21]. There, computing the STA/LTA on the spectral energy, calculated within sliding windows of 4.1 s length, gave more robust results than in the time domain. We hence apply the generalized trigger described previously to a sliding-window spectral representation of the signal (see also the next section).

E. Feature Extraction

Features were selected on the basis of a precursor study [21], according to existing knowledge, general-purpose sound processing, computational considerations, and, again, similarity to the current system. In contrast to the detection and classification stages, no parameters or other specifics of the feature extraction procedure were tuned within the overall joint optimization procedure.

In comparison to the IDC system, our representation puts somewhat stronger emphasis on spectral and cepstral features. The latter are known to be well-suited for spotting harmonics from gas bubble oscillations following underwater explosions [18]. Table II lists the features used, grouped into time, frequency, and cepstral domain. For the frequency domain, overlapping segments of 4.1 s are Fourier transformed every second. The spectral features are then obtained as averages over those

TABLE II
FEATURES EXTRACTED FROM DETECTED SIGNALS. NUMBERS IN PARENTHESES GIVE THE TOTAL NUMBER OF FEATURES IN THAT FEATURE GROUP (FOR EXAMPLE AS A RESULT OF BANDWISE SPECTRAL OR CEPSTRAL CALCULATIONS)

Domain	Features	# of features
Temporal	Duration, zero-crossing rate, total and maximum energy, temporal centroid, energy envelope (4), energy statistical moments (28)	37
Spectral	Spectral flatness, flux, roll-off (21), average statistical moments (28)	49
Cepstral	MFCC means and variances (16), average cepstrum statistical moments (4)	20
		Total: 106

extracted from each of the Fourier transformed segments. Some features were extracted after filtering the entire signal through a filter bank (with frequency ranges 1–3, 3–7, 7–15, 15–31, 31–63, and 63–127 Hz). The features extracted once from every frequency band as well as the unfiltered signal are the following: the first four energy statistical moments, first four spectrum statistical moments, spectral flux, roll-off, and flatness. The extraction of the mel-frequency cepstral coefficients (MFCCs) and average cepstrum statistical moments followed an equivalent procedure in the cepstral domain.

F. Classifiers

In total, we evaluated six different learning machines, four of them SVM classifiers with different kernel functions (see the following) and/or optimization objectives. The other two, LDA and NN classification, can be seen as baseline comparison methods, complementing results from previous studies. Of the SVM classifiers, one used an RBF kernel, one used a linear kernel, one used an ARD kernel, and the last one used again an RBF kernel but with the cross-validation area under the ROC curve (see the following) as the optimization criterion for the overall optimization run instead of the cross-validation classification error. In the following, we briefly reiterate the general problem of supervised learning and classification, as well as each classifier used.

1) Supervised learning and regularized risk minimization: In the supervised learning scenario, adaptation (or training) of a classifier is driven by sample data S (see Section III-C). Let $S=\{(x_i,y_i)|1\leq i\leq \ell\}$ be drawn from an unknown distribution p over $X\times Y$, with X being the input set (corresponding to the feature space $\mathbb{R}^N$ in common classification problems like the present one) and Y being the output set (here, the binary label set $\{-1,+1\}$). Formally, the goal of binary classification is to infer a hypothesis $h: X\rightarrow Y$ that minimizes the expected risk $R_p(h)$

$$R_p(h)=\int_{X\times Y} L_{0-1}\left(y,h(x)\right)\, dp(x,y) \tag{3}$$

where $L_{0-1}(y,z)$ is 0 if $y=z$ and 1 otherwise (i.e., the zero-one loss). Simply put, the classifier should make as few mistakes as possible in guessing an input's label when considering the problem's overall distribution. The loss function need not be symmetric. It can be reasonable to choose a different loss function with $L(0,1)\neq L(1,0)$ to emphasize, for instance, that overlooking an explosive-like event is worse than a false alarm.

As p is unknown, the expected risk in practice has to be estimated using the only known manifestation of p—namely, the sample data S. Different paradigms exist in the literature on how to abstract from a performance measure on S to a valuation of the performance of h on the entire distribution p. One common and successful paradigm is that of *regularized risk minimization* (RRM) [27]. RRM formalizes the intuitive strategy of looking for a hypothesis that is both "simple" and performing well on the training data. Congruence with the training set alone is not a good indicator of a hypothesis' generalization capability to all of p, as, for example, the input-label distribution of S is exactly reproducible if the hypothesis function is only complex enough (i.e., in any sufficiently powerful hypothesis space). Phenomena of a hypothesis representing idiosyncrasies of S instead of properties of p are commonly referred to as overfitting. To avoid overfitting on S, RRM adds a regularization term to the learning objective that penalizes complicated solutions. Therefore, the hypothesis spaces considered are understood to be endowed with a (semi-)norm $\|.\|_{\mathcal{H}}$, serving as a measure of complexity of a hypothesis. The preference for a simple but well-performing hypothesis h is then expressed by minimizing the regularized risk $\mathcal{P}_S$

$$\mathcal{P}_S(\hat{h})=\|\hat{h}\|_{\mathcal{H}}+C\sum_{i=1}^{\ell} L\left(y_i,\hat{h}(x_i)\right). \tag{4}$$

Here, $C>0$ is the so-called regularization parameter, balancing preference for low training error (right summand) against keeping the hypothesis simple (left summand), where, and because, complexity is assumed to correlate to the norm in $\mathcal{H}$. Within the aforementioned framework of RRM, different learning methods can be realized by different choices for the loss L and the admissible hypothesis class $\mathcal{H}$.

2) SVM classification: In their canonical form, SVMs are learning machines for two-class classification tasks on arbitrary input sets, i.e., $Y=\{-1,+1\}$ and X can be any set on which a suitable *kernel function* can be defined (see the following). SVMs utilize as loss function L in (4) a convex relaxation of the zero-one loss to facilitate optimization, namely, the hinge loss $L_{\text{hinge}}(y,f(x))=\max\{0,1-yf(x)\}$. Second, SVMs obtain their hypothesis space by applying the so-called kernel trick [23] to all linear expansions of the input data points. In detail, consider a positive semidefinite (Mercer) kernel function $K: X\times X\rightarrow\mathbb{R}$ [28]. Then, SVMs build on the feature space $\mathcal{H}_K=\text{span}\{K(x,\cdot)|x\in S\}$ and the function class $\mathcal{H}_K^b=\{f=g+b|g\in\mathcal{H}_K, b\in\mathbb{R}\}$. The decision boundary between the two classes is induced by the sign of a function

$f \in \mathcal{H}_K^b$ and is a hyperplane in $\mathcal{H}_K$. Applying the hinge loss and kernelized hypothesis space to the optimization problem of RRM (4) gives the quadratic optimization problem of 1-norm binary soft margin SVMs

$$\underset{f \in \mathcal{H}_K^b}{\text{minimize}} \, \|f\|_K^2 + C \sum_{i=1}^{\ell} L_{\text{hinge}}\left(y_i, f(x_i)\right). \tag{5}$$

In summary, the parameter $C > 0$ controls the tradeoff between reducing the empirical loss L_{hinge} and the complexity of the hypothesis, as measured by its (semi-)norm $\|.\|_K$ in the kernel-induced feature space $\mathcal{H}_K^b$ (where the norm is transferred from $\mathcal{H}_K$ to $\mathcal{H}_K^b$ as a seminorm).

In general, the kernel function K is a crucial element in SVM classification. Its choice, as well as that of C, needs to be made prior and in addition to solving the subsequent optimization problem (5). Examples of common kernel functions, all of which were also used in this study, are the simple linear kernel $K(x, z) = \langle x, z \rangle$, which yields decision boundaries linear in the input space; the arguably most often used RBF kernel $K(x, z) = e^{-\gamma \|x - z\|^2}$, which introduces the bandwidth parameter $\gamma > 0$ as a single free parameter; and the ARD kernel $K(x, z) = e^{-\sum_i \gamma_i (x_i - z_i)^2}$, where i, $0 < i < m$, indexes the m feature dimensions of the input data. The ARD kernel owes its name to the fact that the learned values for its m parameters $\gamma_i > 0$ can provide insight into the relevance of individual features for classification. The regularization parameter C plus any additional kernel parameters are the so-called hyperparameters, which need external tuning or optimization in addition to the training phase itself. This process is referred to as model selection. While the ARD kernel introduces as many free parameters as there are input space dimensions, efficient parameter optimization for the ARD kernel and generalizations thereof exist [29], [30].

3) Baseline algorithms: As baseline methods, we considered LDA and NN classification [16], [31]. We optimized the number k of NNs through cross-validation jointly with the trigger parameters in the same way as for the SVM hyperparameters (see the following). LDA in its basic form has no hyperparameters, and optimization applies here to the trigger parameters only.

4) Data imbalance: In IDC routine operations, there is a clear imbalance between the frequencies of explosive-like and nonexplosive-like hydroacoustic events and detections, with the former occurring at a much scarcer rate. This would give rise to a highly imbalanced classification problem, for which application of the class-insensitive loss of objective (5) may lead to reduced relative accuracy for samples of the underrepresented class. Imbalanced data problems occur widely across application domains, and an extensive array of mitigation strategies exists (see [32] for a comprehensive review and also [33] and [34]). The two canonical types of mitigation strategy intervene either on the data side (e.g., by resampling of the data set) or algorithm side (e.g., by modifying the learner's loss or cost function). For the present hydroacoustic application task, a decision for undersampling of the nonexplosive class to create a close-to-balanced data set was made by IDC early during data set compilation, before any application studies commenced. This resulting almost-balanced IDC data set was described in Section III-C. As also stated there, all events included in the data set had triggered a detection in the IDC's original processing system.

Our strategy here is to apply standard classifiers to the given data set, which is not sampled i.i.d. from the underlying distribution. The oversampling of the explosive-like class reflects the importance of a high *true positive rate (TPR)* (i.e., a high detection rate of explosive-like events). The degree of oversampling is to a certain extent arbitrary. Therefore, we focus on the area under the ROC curve ([17], AUC) as important evaluation criterion (both in the sense of final performance measure and, in one optimization run as an objective function for model selection). To this end, we consider a graded output of each classifier (in contrast to just binary decisions). For SVMs, we take the real-valued output of the decision function f; for LDA, we consider the predicted probability to belong to the positive class; and for k-NN, we use the most basic approach and consider the fraction of the k nearest neighbors voting for the positive class. This continuous output is mapped to a $\{0,1\}$-classification decision by comparing it to a decision threshold t. A ROC graph plots the TPR against the *false alarm rate* (i.e., the fraction of nonexplosive events classified as explosive-like) of the classifier when varying t. This curve reflects what kind of tradeoffs between TPR and false alarm rate the classifier can realize by adjusting t. This enables the tradeoff to be adjusted *after* the classifier has been constructed, which is highly desirable for the present application case. The AUC is the area under the ROC graph, and the AUC value *is independent of the proportion of positive to negative instances in the test set* [17], which makes it ideal for evaluating the classifiers in this study.

G. Processing Chain Optimization

As outlined previously, we view the entire processing chain from detection to classification as one algorithm the parameters of which are to be optimized. We advocate the use of one single objective function for the entire optimization process, here the cross-validation classification error. We also explore the cross-validation area under the ROC curve and examine if this translates to improved ROC curves in the final classifier.

1) Data partitioning and repeated cross-validation: For the cross-validation procedures, we randomly split the available data into five class-stratified *outer* folds (see, e.g., [16, Sec. 7.10.1] for a discussion of the choice of the number of folds). Four of these five folds were used for model selection and training, and the fifth, otherwise unused, test fold was used for the final evaluation of the respective classifier. This was repeated for five times with each outer fold serving once as otherwise unused test partition. This fivefold *outer* cross-validation error calculation was then repeated for a total of ten times on ten different random realizations of outer partitionings. This procedure is known as *repeated cross-validation* (see, e.g., [35]). Note that the 50 resulting performance scores obtained through this repeated cross-validation are not fully statistically independent. Still, they were viewed and evaluated as if they had been 50 independent trials (see Section IV), slightly violating the assumptions underlying the follow-up statistical significance testing.

For those classification algorithms requiring model selection, additional *inner* fivefold cross-validation partitions were defined, splitting each training set. Only the outer procedure used repeated cross-validation.

2) Trigger parameters: As detailed in Section III-D, we used a grid of 7500 trigger parameter combinations as resulting from Table I. For each combination, its corresponding trigger was applied to all raw waveform snippets (which, as described previously, each contain a segment of data having triggered a detection in the current processing system, embedded in ambient noise). Each different trigger parameter combination selects different waveform segments from the raw waveform data, and will thus lead to different values being extracted for each of the features listed in Table II.

3) Treatment of false negatives: Some trigger parameter combinations may not yield a detection on some of the waveform snippets that constitute the data set. Our treatment of these nondetections depends on whether the example lies in the training or test partition for the current evaluation run. If one or more false negatives by the trigger were observed within the training set (i.e., the trigger failed to detect an *explosive-like* signal), that trigger parameter combination was not admitted to competing for the best overall trigger and hyperparameter parameters; in fact, cross-validation performance scores were not even determined. This rationale is driven by the fact that the CTBTO's mission requires special care that no explosive-like signals are missed by the classification algorithm. An additional view is analogue to typical requirements in cascaded classifier design [36], with the trigger algorithm functioning as primary classification stage. If a trigger did not detect a *nonexplosive* signal in a training set, the only consequence was that the training data for the subsequent classifier comprised one less training example for this trigger parameter combination. In the second case of one or more nondetections by the trigger in the test partition, this did not have any consequences until the very last stage of performance score evaluation on the otherwise unused test partition. There, each false negative by the trigger was counted in the same way as a false negative by the classification algorithm, i.e., as a classification error. Likewise, a nondetection by the trigger of a nonexplosive signal was regarded as correct classification.

4) Further preprocessing: The training set was normalized to have zero mean and unit variance. The corresponding transformation was applied to the test data. For all classifiers using RBF kernels, a starting grid value for the bandwidth parameter γ_0 was determined *from the training set* using Jaakkola's heuristic [37].

5) SVM hyperparameter search: For each of the trigger parameter combinations described previously, a joint SVM hyperparameter search was carried out, thus mutually optimizing a cross-validation-based performance measure over both trigger and classifier parameters. For the linear kernel, we probed for each trigger parameter combination a linear "grid" for the regularization parameter C with values of $e^{-7+i*0.32}$, where $i \in \{0, \ldots, 50\}$. For all RBF kernels, the 2-D grid used values of $e^{-7+i*0.8}$, $i \in \{0, \ldots, 20\}$, for C, and $e^{\ln \gamma_0 - 5 + i*0.5}$, $i \in \{0, \ldots, 20\}$, for γ, where the base bandwidth value γ_0 was determined by Jaakkola's heuristic as stated previously. Thus, for each trigger parameter combination, 50 SVM parameters were probed for the linear kernel, and 400 combinations were probed for the RBF kernel.

Once the best-performing combination of trigger parameters and SVM hyperparameters according to the cross-validation error on the training set was found, this combination was either noted and used in the subsequent final evaluation on the so far unused test set. Or, in case of the SVMs using RBF kernels, a second, finer grid search for the bandwidth parameter γ and regularization parameter C was conducted (again on the training set). This is motivated by the fact that the relatively large number of trigger parameter combinations tested required somewhat coarser grid values for the SVM hyperparameters during the mutual search. A second, refining grid on the SVM hyperparameters only thus corresponds to a nested grid search with decreasing search dimensionality, where the first stage serves to fix the trigger parameters and identify a region of stronger interest for the SVM hyperparameters. Letting $\hat{\gamma}$ and $\hat{C}$ denote the best-performing SVM hyperparameters of the joint grid over trigger parameters and hyperparameters, then the second, refined, and also exponentially spaced grid for only the SVM hyperparameters used values of $e^{\ln \hat{\gamma} - 3 + i*0.15}$, where $i \in \{0, \ldots, 40\}$ for γ, and values of $e^{\ln \hat{C} - 3 + i*0.15}$ for C, where $i \in \{0, \ldots, 40\}$.

For all grid runs—both the primary and, if applicable, the secondary, finer, and lower dimensional grid—it was verified that the best SVM hyperparameters did not lie on the grid border. This checks that the overall grid area was initially chosen large enough (i.e., that the cross-validation error landscape does not slope toward better values beyond the grid borders). For the trigger parameters, however, the value ranges are for some parameters bounded by construction and can also contain as few as three discrete values. For these reasons, boundary checks did not extend to trigger parameters.

For the feature sets used in this study, the ARD kernel has 106 parameters (see Section III-E, Table II), making direct parameter search computationally prohibitive. Instead, gradient-based SVM hyperparameter optimization on a maximum-likelihood-based model selection criterion was employed for each trigger parameter combination according to the procedure and setup described in [29]. That approach relies on cross-validation as well, and the same folds as for the other classifiers were used.

Once the best overall trigger parameters and, where applicable, SVM hyperparameters or number of nearest neighbors k were determined, the performance of the corresponding overall ensemble was evaluated on the previously unused test set (and false negatives by the trigger algorithm counted as classification errors as detailed previously).

All machine learning experiments were implemented using the Shark machine learning library[1] [38], [39]. All SVM experiments employed 1-norm binary SVMs with bias term [23].

6) ROC analysis and optimization: Tuning and evaluation of ROC curves were addressed in two ways. For all classifiers, ROC curves were constructed from the classifier's decision function scores obtained on the previously unused test set.

[1] Available for download from http://image.diku.dk/shark/.

TABLE III
EVALUATION OF THE SIX CLASSIFIERS APPLIED IN THIS STUDY (SEE TEXT FOR DESCRIPTIONS). DIAGONAL ENTRIES SHOW MEAN CLASSIFICATION ERRORS (UPPER ROWS) AND MEAN ROC-AUCS (LOWER ROWS) IN PERCENT AVERAGED OVER 50 VALUES OBTAINED THROUGH 10 REPEATED RUNS OF OUTER FIVEFOLD CROSS-VALIDATION. OFF-DIAGONAL ENTRIES SHOW SIGNIFICANCES OF PAIRWISE TWO-SIDED WILCOXON SIGNED-RANK TESTS WITHOUT CORRECTION FOR MULTIPLE COMPARISONS (SEE TEXT FOR RESULTS OF CORRECTIONS). THE SYMBOLS $<$ OR $>$ INDICATE THAT THE ROW ENTRY HAS A SMALLER OR LARGER VALUE, RESPECTIVELY, THAN THE COLUMN ENTRY AT A SIGNIFICANCE LEVEL $p < 0.05$, AND CORRESPONDINGLY, $\ll$ OR $\gg$ DENOTE A TIGHTER SIGNIFICANCE LEVEL OF $p < 0.01$. THE SYMBOL $-$ DENOTES NO DIFFERENCE AT THESE SIGNIFICANCE LEVELS

[%]	svm-rbf	svm-rbf-auc	svm-lin	LDA	k-NN	svm-ard
svm-rbf	3.5 ± 0.6	–	–	$\ll$	$<$	$\ll$
	99.0 ± 0.4	–	–	$\gg$	$\gg$	$\gg$
svm-rbf-auc		3.3 ± 0.3	$\ll$	$\ll$	$\ll$	$\ll$
		99.2 ± 0.3	$>$	$\gg$	$\gg$	$\gg$
svm-lin			3.9 ± 0.3	$\ll$	$\ll$	$\ll$
			98.7 ± 0.3	–	$\gg$	–
LDA				4.2 ± 0.2	–	$\ll$
				98.5 ± 0.4	–	–
k-NN					4.5 ± 0.6	–
					98.0 ± 0.6	–
svm-ard						5.1 ± 0.4
						98.4 ± 0.4

Furthermore, for an RBF kernel, we in an additional run also selected the best trigger parameters and SVM hyperparameters according to the maximum area under the ROC curve (ROC-AUC) as obtained during cross-validation on the training set, instead of using the cross-validation classification error for parameter optimization. In other words, trigger and SVM hyperparameters were deemed fit for final classification if they yielded high ROC-AUC values on the validation fold of the training partition. This classifier (termed svm-rbf-auc) was, in turn, also characterized by ROC analysis on the decision function scores on the previously unused test set.

IV. RESULTS

Table III lists the average test classification errors and ROC-AUC values obtained by the evaluation procedure described previously. All values are averages over 50 values obtained through 10 repeated runs of outer fivefold cross-validation as laid out in Section III-G. As noted previously, the values obtained by repeated cross-validation are not fully independent, slightly violating the assumptions for common statistical hypothesis testing. Keeping this in mind, Table III also shows the results of comparing the performances between classifiers by pairwise nonparametric significance testing (two-sided Wilcoxon signed-rank test), carried out *as if* the performances on the different partitions were statistically independent.

The first two entries in Table III, denoted svm-rbf and svm-rbf-auc, are SVM classifiers using an RBF kernel. For svm-rbf, the trigger parameters and hyperparameters were optimized according to minimization of the cross-validation classification error, and for svm-rbf-auc according to the maximization of the cross-validation area under the ROC curve. Both approaches yield similar performances; while the latter obtains better performance values, the differences between both are not significant. The third entry, svm-lin, denotes an SVM with linear kernel, which performs somewhat worse. The fourth and fifth entries are the two baseline comparison approaches LDA and k-NN (see Section III-F). Both perform again somewhat worse than the linear SVM and comparatively similar to each other, with the k-NN classifier yielding the worst AUC of all classifiers (which can be explained by the coarse interpretation of the k-NN output for ROC computation).

The last classifier, svm-ard, is an SVM utilizing the previously introduced ARD kernel with one kernel parameter for every input feature dimension. This classifier exhibits the significantly worst classification error. This is most likely due to there being only a few times as many training examples in the data set as the ARD kernel has hyperparameters.

In summary, the significance levels in Table III show no significant differences between the two best-performing approaches svm-rbf and svm-rbf-auc. These two, in turn, significantly differentiate themselves from the two baseline approaches LDA and k-NN. All approaches perform better (and almost all significantly so) than the SVM with ARD kernel in terms of classification error and likewise better than the k-NN classifier in terms of AUC value. These tests are not yet corrected for multiple comparisons (i.e., do not take into account that, with an increasing number of pairwise comparisons, there is an increasing probability that one or more discoveries are false).

When correcting via Holm–Bonferroni correction [40] the significance levels from Table III for multiple comparisons (i.e., for the fact that 15 pairwise comparisons were conducted between the 6 classifiers), these findings stay qualitatively correct: almost all differences between classification error values significant at the 1% level remained significant at a weaker significance level of 5%, except that between the svm-rbf and LDA (results not shown). Similarly, when Holm–Bonferroni-correcting significance levels for the AUC values in Table III, differences significant at the 1% level remained significant at the weaker significance level of 5%, with the exception of that between svm-rbf and svm-ard (results not shown).

Table III also lists standard deviations for the error and AUC values: the highest standard deviations were exhibited by the svm-rbf and k-NN approaches with up to 0.6%, which informs the reduced significance level for the comparison between these two methods.[2]

[2]For completeness, we also report the hyperparameter values that would result from hyperparameter selection with svm-rbf-auc using all of the available

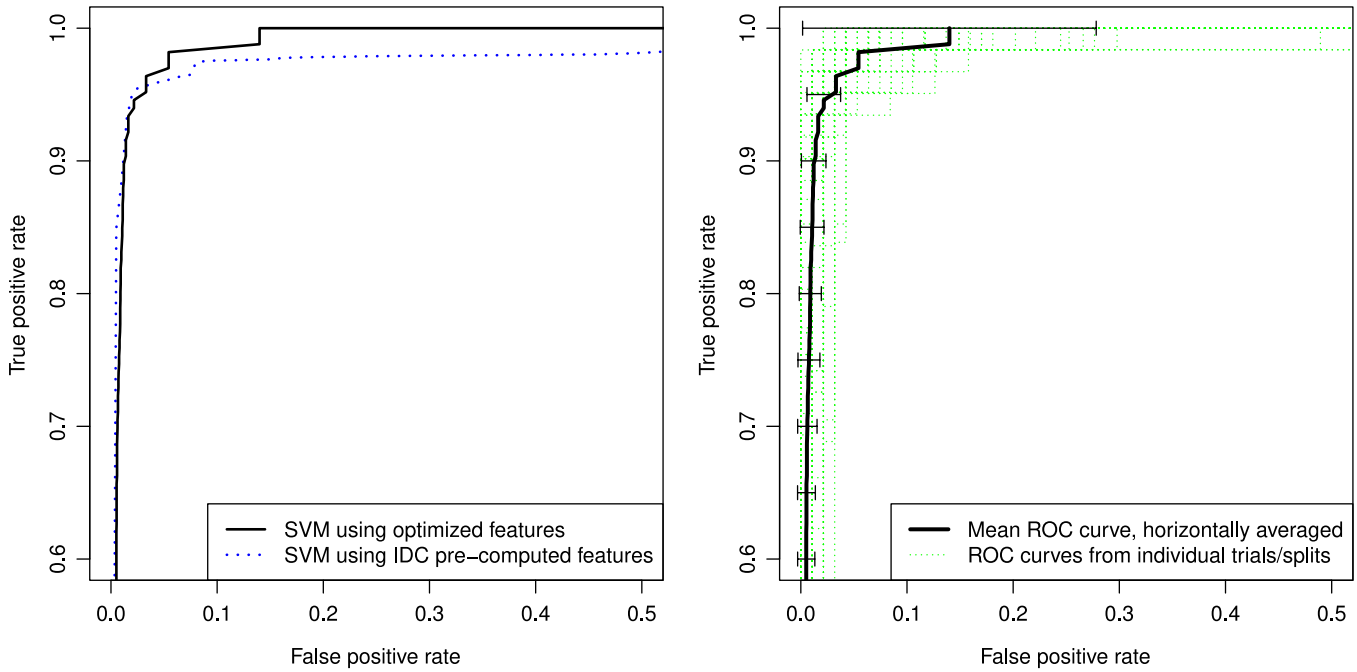

Fig. 2. **Left subplot**: ROC curve of the best-performing approach svm-rbf-auc (solid line in both plots) compared to that of one previous state-of-the-art approach (svm-miss, dotted line). Curves were obtained by horizontal averaging over individual trials, although on a different number of trials (see text). **Right subplot**: the solid line (svm-rbf-auc) is repeated from the left; overlaid, ROC curves of all 50 individual subtrials that were used in the averaging are shown. The proposed classifier reaches full sensitivity at much higher specificity values than svm-miss, which is desired in the application scenario. Also note the cutoff on both axes.

We next review the outcomes established in previous work, comparing to the results from Table III. The rule-based expert system developed by IDC reaches classification errors of around 10% [41]. A naive comparison baseline study also initiated by the authors of this paper used the same raw signal-with-noise snippets as in this study. There, the waveform segments were only minimally preprocessed by trivial filtering and smoothing, before passing raw waveform segments of equal length (which were obtained by naive clipping, thus lacking a proper trigger algorithm altogether) directly to an SVM. This baseline classification method, circumventing as well as possible a trigger and feature extraction stage, reached a classification error of 8.6% (results not shown). For the hydroacoustic binary classification task, the two approaches presented in [3] and [2] achieve classification errors (depending on the exact subapproach taken) between 6.5% and 4.0%, and between 4.9% to 4.3%, respectively. The best approaches in Table III of the present study reach classification errors of 3.3% and 3.5%. In summary, applying state-of-the-art machine learning methods and additionally tailoring them to the application problem at hand successively lowered the classification error for this hydroacoustic application problem from 10% to above or around 4% in previous studies [2], [3] and to 3.3%–3.5% in the current study.

It should be noted that the previous approaches used one trial of fivefold outer cross-validation instead of ten as in the present study. In addition, not all individual subresults of fivefold outer cross-validation runs were available for comparison in nonaveraged form. Hence, statistical significance testing was limited to nonparametric location tests. For completeness, we performed one-sample two-sided Wilcoxon signed-rank location tests, which indicated significant differences (at the 1% level) of the two SVMs with RBF kernels from this study vis-a-vis previously reported methods when not correcting for multiple comparisons. After correction for multiple comparisons (Holm–Bonferroni correction), differences remained significant at the 5% level.

In overall summary, the best results were achieved by the two SVM classifiers using an RBF kernel from this study. While the variant optimizing for a low AUC reached slightly better performance values, we again note that the differences were not significant. This matches our general experience that optimizing for high AUC does not necessarily imply better test AUC values when compared to canonical optimization for low cross-validation error. We overall interpret the results as showing both approaches to perform at equal level while improving on the SVMs with linear and ARD kernel, the baseline algorithms LDA and k-NN, and the reference approaches established in previous work [2], [3], which both used highly task-specific machine learning methods to account for missing values in IDC-extracted feature sets.

Fig. 2 shows a comparison of ROC curves from the best-performing approach of the current study (svm-rbf-auc) and one of the previous state-of-the-art approaches (svm-miss) for which ROC data was available [2]. The two curves in the left subplot represent horizontal averages over the individual trials conducted (although over 50 individual runs for svm-rbf-auc and five individual runs for svm-miss, see previous discussion; also note the cutoff on both axes). The proposed svm-rbf-auc

data: $N_s = 10$ s, $N_\ell = 60$ s, $N_d = 20$ s, $r_S = 1.1$, $r_R = 0.0$, $N_a = N_z = 10$ s, $\gamma = 0.00627$, and $C = 12.807$.

classifier exhibits 100% sensitivity already at higher specificity values than the previous approach svm-miss, which is desired in the hydroacoustic classification task. In particular, the approach svm-miss reaches full sensitivity only at rather low specificity values of around 0.2 (outside the plot axes).

Comparing to previously established results, those of the present study highlight, for one, the relevance of high-quality representation of the respective samples to be classified: improved features can inform common baseline algorithms to perform at par with highly specialized algorithms on less informative features. At the same time, it is remarkable how well an SVM is able to perform on raw waveform data with intentionally close to no preprocessing.

V. Conclusion And Future Work

The interest in machine learning for IMS data processing has notably risen over the last years. This is, among others, caused by a steady increase in certified sensor stations as the IMS nears its completion. Second, as IDC processing rules matured over more than a decade of operations, collaborative studies can explore the potential benefit of incorporating advances made in adaptive processing and pattern recognition since the general inception of the IDC's processing system.

We showed that for the CTBTO hydroacoustic signal classification task, our approach of constructing and jointly optimizing a generic processing pipeline is able to significantly improve on the current state of the art while at the same time remaining structurally similar to the one currently in IDC operations. Our proposed approach of optimizing aspects of preprocessing together with the classifier may, in addition, serve as general blueprint for similar problem settings of combined signal preprocessing and classification.

Testing the proposed framework on longer streams of continuous data, particularly to examine the generalization behavior and robustness of the selected trigger algorithm, as well as its interplay with downstream processing phases, would be the next steps. Independent of this application task, it would be interesting to jointly optimize more powerful signal representation techniques together with the detectors and classifiers operating on them (see, e.g., [6], [19], and [20]).

In overall conclusion, the present work is one successful example of how mutual optimization of several processing stages can make pattern recognition algorithms perform better in practical application tasks.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments and the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO) Preparatory Commission for data provision through its virtual Data Exploitation Centre (vDEC, www.ctbto.org/specials/vdec). M. Tuma would like to thank the German National Academic Foundation, the CTBTO Preparatory Commission, and the Ruhr-University Bochum Research School for their respective support, as well as F. Schiller for the insightful discussions on methodology. C. Igel acknowledges support from the European Commission through project AKMI (PCIG10-GA-2011-303655). All views expressed in this contribution are solely those of the authors and do not necessarily reflect the views of the CTBTO Preparatory Commission.

References

[1] R. Urick, *Principles of Underwater Sound*, 3rd ed. Los Altos, CA, USA: Peninsula, 1996.

[2] M. Tuma, C. Igel, and M. Prior, "Hydroacoustic signal classification using support vector machines," in *Signal and Image Processing for Remote Sensing*, C. Chen, Ed., 2nd ed. Boca Raton, FL, USA: CRC Press, 2012, pp. 37–56.

[3] S. Tschiatschek, N. Mutsam, and F. Pernkopf, "Handling missing features in maximum margin Bayesian network classifiers," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2012, pp. 521–532.

[4] A. Trnkoczy, "Understanding and parameter setting of STA/LTA trigger algorithm," in *New Manual of Seismological Observatory Practice (NMSOP-2)*, P. Bormann, Ed. Potsdam, Germany: IASPEI, 2012.

[5] M. Withers *et al.*, "A comparison of select trigger algorithms for automated global seismic phase and event detection," *Bull. Seismol. Soc. Amer.*, vol. 88, no. 1, pp. 95–106, 1998.

[6] M. Davy, A. Gretton, A. Doucet, and P. Rayner, "Optimized support vector machines for nonstationary signal classification," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 442–445, Dec. 2002.

[7] D. Hafemeister, "The comprehensive test ban treaty: Effectively verifiable," *Arms Control Today*, vol. 38, pp. 6–12, 2008.

[8] "Comprehensive Nuclear-Test-Ban Treaty," UN Gen. Assem., New York, NY, USA, Sep. 10, 1996. [Online]. Available: www.ctbto.org/fileadmin/content/treaty/treatytext.tt.html

[9] R. Roberts, A. Christoffersson, and F. Cassidy, "Real-time event detection, phase identification and source location estimation using single station three-component seismic data," *Geophys. J. Int.*, vol. 97, no. 3, pp. 471–480, 1989.

[10] N. Arora, S. Russell, and E. Sudderth, "NET-VISA: Network processing vertically integrated seismic analysis," *Bull. Seismol. Soc. Amer.*, vol. 103, no. 2A, pp. 709–729, Apr. 2013.

[11] A. Thunborg, *Science for Security. Verifying the Comprehensive Nuclear-Test-Ban Treaty*. Vienna, Austria: CTBTO Preparatory Comm., 2009.

[12] A. Conjares, *CTBT Science and Technology Conference (S&T 2011), Book of Abstracts*. Vienna, Austria: CTBTO Preparatory Comm., 2011.

[13] C. Riggelsen and M. Ohrnberger, "A machine learning approach for improving the detection capabilities at 3C seismic stations," *Pure Appl. Geophys.*, vol. 171, no. 3, pp. 395–411, Mar. 2014.

[14] M. Slinkard *et al.*, "Multistation validation of waveform correlation techniques as applied to broad regional monitoring," *Bull. Seismol. Soc. Amer.*, vol. 104, no. 6, pp. 2768–2781, 2014.

[15] B. Sick and M. Joswig, "Unsupervised clustering of seismic events in an on-site-inspection scenario," in Geophysical Research Abstracts, European Geosciences Union General Assembly, Vienna, Austria, 2012, vol. 14, Paper 13174.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[17] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[18] J. Hanson *et al.*, "Operational processing of hydroacoustics at the prototype international data center," *Pure Appl. Geophys.*, vol. 158, no. 3, pp. 425–456, 2001.

[19] S. Gabarda and G. Cristobal, "Detection of events in seismic time series by time-frequency methods," *IET Signal Process.*, vol. 4, no. 4, pp. 413–420, Aug. 2010.

[20] B. Boashash and P. O'Shea, "A methodology for detection and classification of some underwater acoustic signals using time-frequency analysis techniques," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 38, no. 11, pp. 1829–1841, Nov. 1990.

[21] V. Rørbech, "Processing and Classification of Hydroacoustic Signals for Nuclear Test-Ban-Treaty Verification," M.S. thesis, Faculty Sci., Univ. Copenhagen, København, Denmark, 2011.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[23] B. Schölkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2002.

[24] M. Prior, "A test to identify signals associated with underwater explosions," in Geophysical Research Abstracts, European Geosciences Union General Assembly, Vienna, Austria, 2008, vol. 10, Paper A-03035.

[25] S. Vaidya, R. Engdahl, R. Le Bras, K. Koch, and O. Dahlman, "Strategic initiative in support of CTBT data processing: VDEC (Virtual Data Exploitation Centre)," in *CTBTO International Scientific Studies*. Vienna, Austria: CTBTO Preparatory Comm., 2009.
[26] M. Joswig, "Pattern recognition for earthquake detection," *Bull. Seismol. Soc. Amer.*, vol. 80, pp. 170–186, 1990.
[27] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.
[28] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.
[29] T. Glasmachers and C. Igel, "Maximum likelihood model selection for 1-norm soft margin SVMs with multiple parameters," *IEEE Trans. Pattern Recognit. Mach. Intell.*, vol. 32, no. 8, pp. 1522–1528, Aug. 2010.
[30] T. Glasmachers and C. Igel, "Gradient-based adaptation of general Gaussian kernels," *Neural Comput.*, vol. 17, no. 10, pp. 2099–2105, Oct. 2005.
[31] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[32] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[33] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data Mining and Knowledge Discovery Handbook*, 2nd ed. Berlin, Germany: Springer-Verlag, 2010, pp. 853–867.
[34] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.
[35] J.-H. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Comput. Statist. Data Anal.*, vol. 53, no. 11, pp. 3735–3745, Sep. 2009.
[36] A. Prasoon *et al.*, "Cascaded classifier for large-scale data applied to automatic segmentation of articular cartilage," in *Proc. SPIE, Med. Imaging, Image Process.*, 2012, vol. 8314, pp. 1–9.
[37] T. Jaakkola, M. Diekhaus, and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies," in *Proc. 7th Int. Conf. Intell. Syst. Mol. Biol.*, 1999, pp. 149–158.
[38] C. Igel, T. Glasmachers, and V. Heidrich-Meisner, "Shark," *J. Mach. Learn. Res.*, vol. 9, pp. 993–996, 2008.
[39] T. Glasmachers and C. Igel, "Maximum-gain working set selection for support vector machines," *J. Mach. Learn. Res.*, vol. 7, pp. 1437–1466, 2006.
[40] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian J. Statist.*, vol. 6, pp. 65–70, 1979.
[41] IDC, personal communication, 2009.

Matthias Tuma received the Diploma degree in physics from Ruhr-University Bochum, Bochum, Germany, in 2008, where he is currently working toward the Ph.D. degree in electrical engineering with the Institut für Neuroinformatik.

He has been working on data processing problems related to verification of the Comprehensive Nuclear-Test-Ban Treaty since 2006. His research also focuses on efficient training of online multiclass support vector machines. In 2015, he joined the Joint Planning Staff of the World Climate Research Programme, Geneva, Switzerland, where his work includes science coordination and data standards for climate modelling and observations.

Valdemar Rørbech received the M.S. degree in computer science from the Department of Computer Science, University of Copenhagen, Copenhagen, Denmark, in 2012, with specialization in digital signal processing and pattern recognition.

He is currently a System Architect on complex systems with NetCompany A/S, Copenhagen, where his research interests include classification, active learning, and sound processing, as well as system architecture and mobile applications.

Mark K. Prior received the B.Sc. degree in physics from the University of Birmingham, Birmingham, U.K., in 1988 and the Ph.D. degree in underwater acoustics from the Institute of Sound and Vibration Research, University of Southampton, Southampton, U.K., in 1996.

He studied many aspects of sonar performance modelling while working at the Admiralty Research Establishment, Portland, U.K., and SACLANTCEN (now CMRE), La Spezia, Italy. Between 2007 and 2014, he worked for CTBTO in Vienna, Austria, where he was responsible for automatic signal processing strategies applied to the analysis of underwater-acoustic data gathered for the purposes of global nuclear-test-ban monitoring. He is currently with TNO, The Hague, The Netherlands, researching sonar performance modelling and the impact of anthropogenic sound on the marine environment.

Christian Igel (SM'04) studied computer science at the Technical University of Dortmund, Dortmund, Germany. He received the Doctoral degree from the Faculty of Technology, Bielefeld University, Bielefeld, Germany, in 2002 and the Habilitation degree from the Department of Electrical Engineering and Information Sciences, Ruhr-University Bochum, Bochum, Germany, in 2010.

From 2002 to 2010, he was a Junior Professor for optimization of adaptive systems with the Institut für Neuroinformatik, Ruhr-University Bochum. In October 2010, he was appointed a Professor with special duties in machine learning with DIKU, the Department of Computer Science, University of Copenhagen, Copenhagen, Denmark. Since December 2014 he has been a full professor at DIKU.