Pattern Recognition 43 (2010) 2732–2752

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs

Wael Khreich [a,*], Eric Granger [a], Ali Miri [b], Robert Sabourin [a]

[a] Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), École de technologie supérieure, Université du Québec, 1100 Notre-Dame Ouest, Montreal, QC, Canada
[b] School of Information Technology and Engineering (SITE), University of Ottawa, 161 Louis Pasteur, Ottawa, ON, Canada

ARTICLE INFO

Article history:
Received 5 September 2009
Received in revised form
8 January 2010
Accepted 7 March 2010

Keywords:
Receiver operating characteristics
Combination of classifiers
Limited and imbalanced data
Hidden Markov models
Anomaly detection
Computer and network security

ABSTRACT

Hidden Markov models (HMMs) have been shown to provide a high level performance for detecting anomalies in sequences of system calls to the operating system kernel. Using Boolean conjunction and disjunction functions to combine the responses of multiple HMMs in the ROC space may significantly improve performance over a "single best" HMM. However, these techniques assume that the classifiers are conditional independent, and their of ROC curves are convex. These assumptions are violated in most real-world applications, especially when classifiers are designed using limited and imbalanced training data. In this paper, the iterative Boolean combination (*IBC*) technique is proposed for efficient fusion of the responses from multiple classifiers in the ROC space. It applies all Boolean functions to combine the ROC curves corresponding to multiple classifiers, requires no prior assumptions, and its time complexity is linear with the number of classifiers. The results of computer simulations conducted on both synthetic and real-world host-based intrusion detection data indicate that the *IBC* of responses from multiple HMMs can achieve a significantly higher level of performance than the Boolean conjunction and disjunction combinations, especially when training data are limited and imbalanced. The proposed *IBC* is general in that it can be employed to combine diverse responses of any crisp or soft one- or two-class classifiers, and for wide range of application domains.

1. Introduction

Intrusion detection systems (IDS) is used to identify, assess, and report unauthorized computer or network activities. Host-based IDSs (HIDS) are designed to monitor the activities of a host system and state, while network-based IDSs (NIDS) monitor the network traffic for multiple hosts. HIDSs and NIDSs have been designed to perform misuse detection and anomaly detection. Anomaly-based intrusion detection allows to detect novel attacks for which the signatures have not yet been extracted [1]. In practice, anomaly detectors will typically generate false alarms due in large part to the limited data used for training, and to the complexity of underlying data distributions that may change dynamically over time. Since it is very difficult to collect and label representative data to design and validate an anomaly detection systems (ADS), its internal model of normal behavior will tend to diverge from the underlying data distribution.

In HIDSs applied to anomaly detection, operating system events are usually monitored. Since system calls are the gateway between user and kernel mode, traditional host-based anomaly detection systems monitor deviation in system call sequences. Forrest et al. [2] confirmed that short sequences of system calls are consistent with normal operation, and unusual burst will occur during an attack. Their anomaly detection system, called sequence time-delay embedding (STIDE), is based on look-up tables of memorized normal sequences. During operations, STIDE must compare each input sequence to all "normal" training sequences. The number of comparisons increases exponentially with the detector window size. Moreover, STIDE is often used for design and validation of other state-of-the-art detectors. Various neural and statistical detectors have been applied to learn the normal process behavior through system call sequences [3]. Among these, techniques based on discrete hidden Markov models (HMMs) [4] have been shown to produce a very high level of performance [3]. A well trained HMM is able to capture the underlying structure of the monitored application and detect deviations from "normal" system call sequences. Once trained, an HMM provides a fast and compact detector, with tolerance to noise and uncertainty.

Designing an HMM for anomaly detection involves estimating HMM parameters and order (number of hidden states, N) from the training data. The value of N has a considerable impact not only on

* Corresponding author.
E-mail addresses: khreichwael@yahoo.ca, wael.khreich@livia.etsmtl.ca (W. Khreich), eric.granger@etsmtl.ca (E. Granger), samiri@site.uottawa.ca (A. Miri), robert.sabourin@etsmtl.ca (R. Sabourin).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.03.006

the detection rate but also on the training time. In the literature on HMMs applied to anomaly detection [3,5,6], the number of states is often chosen heuristically or empirically using validation data.[1] In addition, HMMs are designed to provide accurate results for a particular window size and anomaly size. Therefore, a single best HMM will not provide a high level of performance over the entire detection space. In a previous work [9], the authors proposed a multiple-HMM (μ-HMM) approach, where each HMM is trained using a different number of hidden states. During operations, each HMM outputs a probability that the HMM produced the input sequence. HMM responses are combined in the receiver operating characteristics (ROC) space according to the maximum realizable ROC (MRROC) technique [10]. This technique is robust in imprecise environments where for instance prior probabilities and/or classification costs may change [11], and can be always applied in practice without any assumption of independence between detectors [10]. Results have shown that this μ-HMMs approach can provide a significant increase in performance over a single best HMM and STIDE [9].

Boolean functions—especially the conjunction AND and disjunction OR operations—have recently been investigated to combine crisp or soft detectors within the ROC space. Successful applications for such combination include machine learning [12,13], biometrics [14], bio-informatics [15,16], automatic target recognition [17,18]. Boolean combination (BC) based on conjunction or disjunction has been shown to improve performance over the MRROC in many applications, yet requires idealistic assumptions in which the detectors are conditionally independent and their respective ROC curves are smooth and proper. In contrast, applying all Boolean functions, using an exhaustive brute-force search to determine optimal combinations leads to an exponential explosion of combinations, which is prohibitive even for a small number of crisp detectors [13].

In practice, neural and statistical classifiers are typically trained with limited amount of representative data, and class distributions are often complex and imbalanced, which leads to concavities on empirical ROC curves, and to poor performance. This issue is especially critical in binary classification problems, such as anomaly detection, where samples from the positive class[2] are inherently rare, and are costly to analyze. In such cases the ROC curve typically comprises large concavities. Some authors have argued that ROC curves can be simply repaired using the ROCCH prior to the conjunction or disjunction combinations [16]. However this technique is inefficient, since the ROCCH selects thresholds of the locally superior points and discard the rest. This leads to a loss of diverse information which could be used to improve the performance.

In this paper, the problem of ROC-based combination is addressed in the general case, where detectors are trained with limited and imbalanced training data. An iterative Boolean combination (IBC_{ALL})[3] technique is proposed to efficiently combine the responses from multiple detectors in the ROC space. In contrast with most techniques in literature, where only the AND or OR are investigated, the IBC_{ALL} exploits *all* Boolean functions applied to the ROC curves, and it does not require any prior assumption regarding the independence of the detectors and the convexity of ROC curves. At each iteration, the IBC_{ALL} selects the combinations that improve the ROCCH and recombines them with the original ROC curves until the ROCCH ceases to improve. Although it seeks a sub-optimal set of combinations, the IBC_{ALL} is very efficient in practice and does not suffer from the exponential explosion [13], and it provides a higher level of performance than related techniques in literature [10,14,16]. The IBC_{ALL} technique can be applied when training data are limited and test data are heavily imbalanced. Another advantage of the proposed technique is its ability to repair the concavities in the ROC curve when applied to combine the responses of the same ROC curve. The IBC_{ALL} is general in the sense that can be applied to combine the responses of any soft, crisp, or hybrid detector in the ROC space, whether the corresponding curves result from the same detector trained on different data or trained according to different parameters, or from different detectors trained on the same data.

During computer simulations, multiple HMMs are applied to anomaly detection based on system calls. The performance obtained by combining the responses of μ-HMMs in ROC space with the proposed IBC_{ALL} technique is compared to that of MRROC fusion [10], to that of the conjunction (BC_{AND}) and disjunction (BC_{OR}) combinations [14,16], and to that of STIDE (for reference). To investigate the effect of repairing the concavities of the ROC curves on the combinations, each ROC curve is first repaired using the IBC_{ALL} technique, and then all curves are combined with the MRROC. This is also compared to the largest concavity repair (*LCR*) proposed in [19]. The impact on performance of using different training set sizes, detector window sizes and anomaly sizes is assessed through the area under the curve (AUC) [20], partial AUC [21], and true positive rate (*tpr*) at a fixed false positive rate (*fpr*). The experiments are conducted on both synthetically generated data and sendmail data from the University of New Mexico (UNM) data sets.

The rest of this paper is organized as follows. The next section describes the application of discrete HMMs in anomaly-based HIDS. In Section 3, existing techniques for combination of detectors in the ROC space are presented. The proposed technique for iterative Boolean combination of detector responses is presented in Section 4. Section 5 presents the experimental methodology (data sets, evaluation methods and performance metrics) used for proof-of-concept computer simulations. Finally, simulation results are presented and discussed in Section 6.

2. Anomaly detection with HMMs

A discrete-time finite-state HMM is a stochastic process determined by the two interrelated mechanisms—a latent Markov chain having a finite number of states, and a set of observation probability distributions, each one associated with a state. At each discrete time instant, the process is assumed to be in a state, and an observation is generated by the probability distribution corresponding to the current state. HMM parameters are usually trained using the Baum–Welch (BW) algorithm [22]—a specialized expectation maximization technique to estimate the parameters of the model from the training data. Theoretical and empirical results have shown that, given an adequate number of states and a sufficiently rich set of observations, HMMs are capable of representing probability distributions corresponding to complex real-world phenomena in terms of simple and compact models, with tolerance to noise and uncertainty. For further details regarding HMM the reader is referred to the extensive literature [4,23].

Formally, a discrete-time finite-state HMM consists of N hidden states in the finite-state space $S=\{S_1,S_2,\ldots,S_N\}$ of the Markov process. Starting from an initial state S_i, determined by the initial state probability distribution π_i, at each discrete-time instant, the

[1] This is also the case in many other applications of ergodic HMMs. However, recently nonparametric Bayesian approaches have been proposed to overcome HMMs order selection [7,8].

[2] The positive or target class is typically the class of interest for the detection problem. For anomaly detection, the target class corresponds to the intrusive or abnormal class.

[3] The subscript "ALL" is used to emphasize that the *IBC* technique employs all Boolean functions.

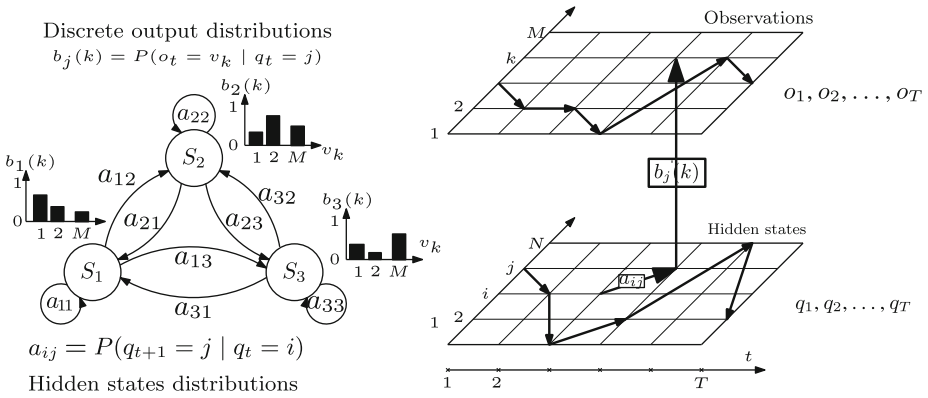

Fig. 1. Illustration of a fully connected three state HMM with a discrete output observations (left). Illustration of a discrete HMM with N states and M symbols switching between the hidden states q_t and generating the observations o_t (right). The state $q_t{=}i$ denotes that the state of the process at time t is S_i.

process transits from state S_i to state S_j according to the transition probability distribution a_{ij} ($1 \le i,j \le N$). As illustrated in Fig. 1, the process then emits a symbol v according to the output probability distribution $b_j(v)$, which may be discrete or continuous, of the current state S_j. The model is therefore parametrized by the set $\lambda = (\pi, A, B)$, where vector $\pi = \{\pi_i\}$ is initial state probability distribution, matrix $A = \{a_{ij}\}$ denotes the state transition probability distribution, and matrix $B = \{b_j(k)\}$ is the state output probability distribution.

Estimating the parameters of a HMM requires the specification of its order (i.e., the number of hidden states N). The value of N may have a significant impact on both detection performance and training time. The time and memory complexity of BW training is $\mathcal{O}(N^2T)$ and $\mathcal{O}(NT)$, respectively, for a sequence of length T symbols. In the literature on HMMs applied to anomaly detection [3,5,6], the number of states is often overlooked and typically chosen heuristically. In addition, a single "best" HMM will not provide a high level of performance over the entire detection space. A multiple-HMMs (μ- HMMs) approach, where each HMM is trained using a different number of hidden states, and where HMM responses are combined according to the MRROC has significantly improved performance over a single best HMM and STIDE [9].

3. Fusion of detectors in the receiver operating characteristic (ROC) space

A *crisp* detector (e.g., STIDE) outputs only a class label and produces a single operational data point in the ROC plane. In contrast, a *soft* detector (e.g., HMM) assigns scores or probabilities to the input samples, which can be converted to a crisp detector by setting a threshold on the scores. Given a detector and a set of test samples, the true positive rate (*tpr*) is the proportion of positives correctly classified over the total number of positive samples. The false positive rate (*fpr*) is the proportion of negatives incorrectly classified (as positives) over the total number of negative samples. A ROC curve is a two-dimensional curve in which the *tpr* is plotted against the *fpr*. A parametric ROC curve typically assumes that a pair of normal distributions underlies the data [24,25] and increases monotonically with the *fpr*. In practice, an empirical ROC curve may be obtained by connecting the observed (*tpr*, *fpr*) pairs of detectors, therefore it makes minimal assumptions [12]. Given two operating points, say a and b, in the ROC space, a is defined as *superior* to b if $fpr_a \le fpr_b$ and $tpr_a \ge tpr_b$. If one ROC curve has all its points superior to those of another curve, it *dominates* the latter. If a ROC curve has $tpr_x > fpr_x$ for all its points x then, it is a *proper* ROC curve. Finally, the area under the ROC curve (AUC) is the fraction of positive–negative pairs that are ranked correctly (see Section 5.3).

The rest of this section provides an overview of techniques in literature for combining detectors and repairing curves within the ROC space. It includes the stochastic interpolation of the maximum realizable ROC (MRROC) or ROCCH, and the conjunction and disjunction rules for combining crisp and soft detectors.

3.1. Maximum realizable ROC (MRROC)

Given that the underlying distributions are generally considered fixed, a parametric ROC curve will always be proper and convex. In practice, an ROC plot is a step-like function which approaches a true curve as the number of samples approaches infinity. An empirical ROC curve is therefore not necessarily convex and proper as illustrated in Fig. 2. Concavities indicate local performance that is worse than random behavior. These concavities occur with unequal variances between classes, with large skew in the data, or when the classifier is unable to capture the modality of the data (e.g., classifying multimodal data with a linear classifier).

As shown in Fig. 2, the ROCCH of an empirical ROC is the piecewise outer envelope connecting only the superior points of an ROC with straight lines. The ROCCH of a single crisp detector connects its resulting point to the (0, 0) and (1, 1) points. The ROCCH is also applicable to multiple detectors which may be crisp or soft. If one detector has a dominant ROC curve over *fpr* values, its convex hull constitutes the overall ROCCH. When there is no clear dominance among multiple detectors across different regions of the ROC space, the ROCCH corresponds to the envelope connecting the superior points in each region (Fig. 2a). The ROCCH formulation has been proven to be robust in imprecise environments [11]. As environmental conditions change, such as prior probabilities and/or costs of errors, only the portion of interest will change, not the hull itself. This change of conditions may lead to shifting the optimal operating point to another threshold or classifier on the convex hull.

The ROCCH is useful for combining detectors. The idea is based on a simple interpolation between the corresponding detectors [10,26]. In practice, this is achieved by randomly alternating detectors responses proportionately between the two corresponding vertices of the line segment on the convex hull where the desired operational point lies. This approach has been called the maximum realizable ROC (MRROC) in [10] since it represents a system that is equal to, or better than, all the existing systems for all Neyman–Pearson criteria. Hereafter, the acronyms ROCCH and MRROC will be used interchangeably. The performance of the composite detector C_c can be readily set to any point along the line

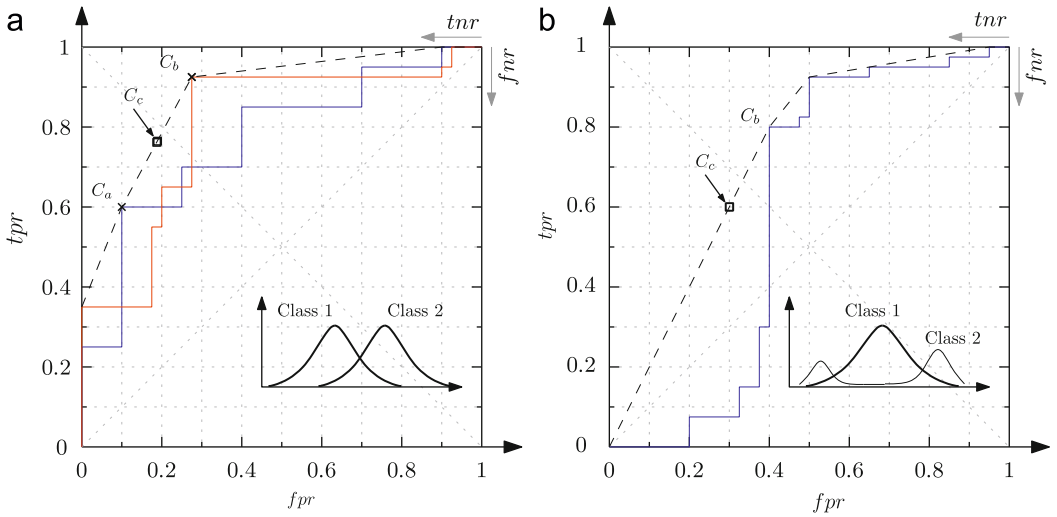

Fig. 2. Illustration of the ROCCH (dashed line) applied to: (a) the combination of two soft detectors (b) the repair of concavities of an improper soft detector. For instance, in (a) the composite detector C_c is realized by randomly selecting the responses from C_a half of the time and the other half from C_b.

segment, connecting C_a and C_b, simply by varying the desired fpr_c and thus the ratio between the number of times one detector is used relative to the other.

The MRROC considers only the responses of detectors that lie on the facet of the ROCCH, since they are potentially "optimal", and discards the responses not touching the ROCCH [10,11]. However, this may degrade the performance due to loss of information since inferior detectors are not exploited for decisions. Indeed, these detectors may contain valuable *diverse* information. As detailed next, some combination techniques have been proposed in literature to improve upon the ROCCH.

3.2. Repairing concavities

Repairing concavities is useful in itself for situations with only one detector with some concavities in its ROC curve. Repairing these concavities could be useful to improve the performance. The MRROC may be used to repair the ROC concavities by discretizing and interpolating between thresholds of superior points. Flach and Wu proposed a technique for largest concavity repairing (*LCR*) [19]. It consists of inverting the largest concavity section with reference to the mid point of the local line segment on the ROCCH. The intuition is that points underneath the ROCCH can be mirrored in the same way negative classifiers (under the ascending diagonal) can be inverted. The *LCR* consist in determining the two thresholds on the ROCCH, limiting the section to be inverted, and then in negating the responses with reference to the mid point of the corresponding line segment of the ROCCH. In some cases the *LCR* may provide a higher level of performance than the MRROC.

3.3. Conjunction and disjunction rules for crisp detectors

The Boolean conjunction (AND) and disjunction (OR) fusion functions were first introduced for combining crisp detectors [27]. Other authors such as Fawcett [12] have noted that it is sometimes possible to find nonlinear combinations of detectors which produce an ROC exceeding their convex hulls.

As illustrated in Fig. 3, the conjunction rule decreases the *fpr* at the expenses of decreasing the *tpr*, thus providing a more conservative performance than each of the original detectors. Analogously, the disjunction rule increases the *tpr* at the expenses of increasing the *fpr*, providing a more aggressive performance than each of the original detectors. When these fusions are

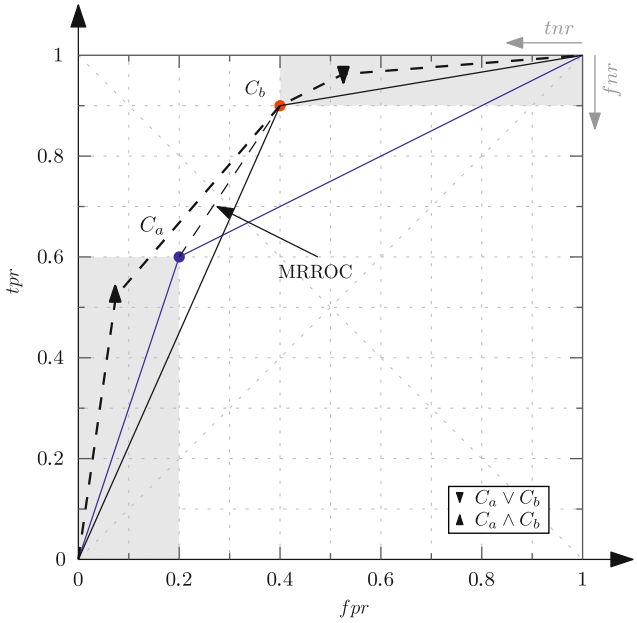

Fig. 3. Examples of combination of two conditionally independent crisp detectors, C_a and C_b, using the AND and OR rules. The performance of the their combination is shown superior to that of the MRROC. The shaded regions are the expected performance of combination when there is an interaction between the detectors.

considered alone—outside the ROC space—their achieved performance may not be of considerable interest as advocated by Daugman in [27]. However, depending on detector interaction, within the ROC space, these fusion rules may produce a new convex hull that is superior than that of existing detectors alone. In addition, the new MRROC curve provides the flexibility of choosing any operating point which lies on its hull by interpolating between the relevant vertices as described previously.

The conditional independence assumption among the detectors simplifies the computation. In this cases, the combination rules depend only on the true and false positive rates. Let (tpr_a, fpr_a) and (tpr_b, fpr_b) be the true positive and false positive rates of detectors C_a and C_b, respectively. Under the conditional assumption, the performance of the composite crisp detectors C_c is given in Table 1 [12,28]. These formulas stem from the conditional independence

Table 1
Combination of conditionally independent detectors.

$P_{11\|1}=tpr_a tpr_b$	$P_{11\|0}=fpr_a fpr_b$
$P_{10\|1}=(1-tpr_a)tpr_b$	$P_{10\|0}=(1-fpr_a)fpr_b$
$P_{01\|1}=tpr_a(1-tpr_b)$	$P_{01\|0}=fpr_a(1-fpr_b)$
$P_{00\|1}=(1-tpr_a)(1-tpr_a)$	$P_{00\|0}=(1-fpr_a)(1-fpr_b)$

Table 2
Combination of conditionally dependent detectors.

$tpr_a tpr_b < P_{11\|1} < \min(tpr_a, tpr_b)$
$P_{10\|1}=tpr_a-P_{11\|1}$
$P_{01\|1}=tpr_b-P_{11\|1}$
$P_{00\|1}=1-tpr_a-tpr_a+P_{11\|1}$
$(1-fpr_a)(1-fpr_b) < P_{00\|0} < \min(1-fpr_a, 1-fpr_b)$
$P_{10\|0}=(1-fpr_b)-P_{00\|0}$
$P_{01\|0}=(1-fpr_a)-P_{00\|0}$
$P_{11\|0}=fpr_a+fpr_b-1+P_{00\|0}$

of probability. For instance, the probability that both detectors correctly classify a positive test sample is given by

$$P_{11|1}=\Pr(C_a=1,C_b=1|1)=\Pr(C_a=1|1)\Pr(C_b=1|1)=tpr_a tpr_b.$$

As mentioned, this assumption is violated in most real-world applications. In the more realistic conditionally dependent case, the performance of the composite crisp detectors C_c is given in Table 2. The performance now depends on the positive ($P_{11|1}$) and negative ($P_{00|0}$) correlations between detectors [28]. That is, the joint distributions of both detectors are required, and the performance can now be anywhere in the shaded regions of Fig. 3.

An attempt to characterize this dependence is given in [29], where the correlation coefficient between the detector scores is shown to be useful for predicting the "best" decision fusion rule, as well as for evaluating the quality of detectors. More recently, in order to avoid the restrictive conditional assumption among detectors, the combination rules were extended to include all Boolean functions [13]. By ranking these combinations according to their likelihood ratios, the optimal rules can be obtained. However, due to the doubly exponential explosion of combinations—for n detectors there is 2^n possible outputs resulting in 2^{2^n} possible combinations—the proposed global search for the optimal rules is impractical.

3.4. Conjunction and disjunction rules for combining soft detectors

Several authors have proposed the application of the Boolean AND and OR fusion functions to combine soft detectors. For a pair-wise combination, the fusion function is applied to each threshold on the first ROC curve with respect to each threshold on of the second curve. The optimum threshold, as well as the combination function, is then found according to the Neyman–Person test [30]. That is for each value of the *fpr*, the point which has the maximum *tpr* value is selected, along with the corresponding thresholds and Boolean function to be used during operations.

Haker et al. [16] proposed to apply the AND and OR functions to combine a pair of soft detectors under the assumption of conditional independence between detectors (Table 1), and when both detectors are proper and convex. The authors proposed a set of "maximum likelihood combination" rules (see Table 3) to select the combination rules or original detectors to be employed. For instance, the AND rule is selected if the first condition ($C_a=1$, $C_b=1$ in Table 3) is exclusively true, while the OR rule is selected when the first three conditions are true. Otherwise one of the individual detectors is selected. This selection may however discard important combinations since for two thresholds several combination rules may emerge. For example, by computing these theoretical conditions for the ROC curves shown in Fig. 3, one can observe that the first and third conditions are verified in Table 3, hence only C_b is considered for these two points. However, as shown in Fig. 3, both the AND and OR combination improve the performance over the original detectors.

Table 3
The maximum likelihood combination of detectors C_a and C_b as proposed in [16].

C_a	C_b	Selection rules for C_c
1	1	$tpr_a tpr_b \geq fpr_a fpr_b$
1	0	$tpr_a(1-tpr_b) \geq fpr_a(1-fpr_b)$
0	1	$(1-tpr_a)tpr_b \geq (1-fpr_a)fpr_b$
0	0	$(1-tpr_a)(1-tpr_b) \geq (1-fpr_a)(1-fpr_b)$

Tao and Veldhuis [14] applied and compared AND versus OR Boolean function separately for combining multiple ROC curves using a pair-wise combination. They showed that the OR rule emerges most of the time in their biometrics application. Oxley et al. [18] proved that, under the independence assumption between detectors, a Boolean algebra of families of classification systems is isomorphic to a Boolean algebra of ROC curves. Shen [31] tried to characterize the effect of correlation on the AND and OR combination rules, using a bivariate normal model. He showed that discrimination power is higher when the correlation is of opposite sign.

Most research has addressed the problem of Boolean combinations under the assumption of smooth, convex and proper ROC curves. Such curves results from a parametric models, or when the data are abundant for both classes. In the ideal case, when both conditional independence and convexity assumptions are fulfilled, the AND and OR combinations are proven to be optimal, providing a higher level of performance than the original ROC curves [13,32,33]. When provided with limited and imbalanced data for training and validation, the ROC curves may be improper and large concavities will appear. When either one of the assumptions is violated, the performance of these combinations will be sub-optimal. In contrast to crisp detectors, the correlation between soft detector decisions also depends on the thresholds selection. At different thresholds on two ROC curves the conditional independence may be violated and different type of correlations may be introduced.

4. A Boolean combination (BC_{ALL}) algorithm for fusion of detectors

4.1. Boolean combination of two ROC curves

In this paper, a general technique for Boolean combination (BC_{ALL}) is proposed for fusion of detector responses in the ROC space. In particular, this algorithm can exploit information from the ROC curves when detectors are trained from limited and imbalanced data. In this case, the ROC curves typically comprise concavities and are not necessarily proper. Fig. 4 presents the block diagram of a system that combines the responses of two HMMs in the ROC space according to the BC_{ALL} technique. It involves fusing responses of detectors using all Boolean functions, prior to applying the MRROC. In contrast with most work in literature, where either AND or OR functions are applied, the BC_{ALL} technique takes advantage of all Boolean functions applied to the ROC curves and selects those that improve the ROCCH.

The main steps of BC_{ALL} are presented in Algorithm 1. The BC_{ALL} technique inputs a pair of ROC curves defined by their decision thresholds, T_a and T_b, and the labels for the validation set. Using

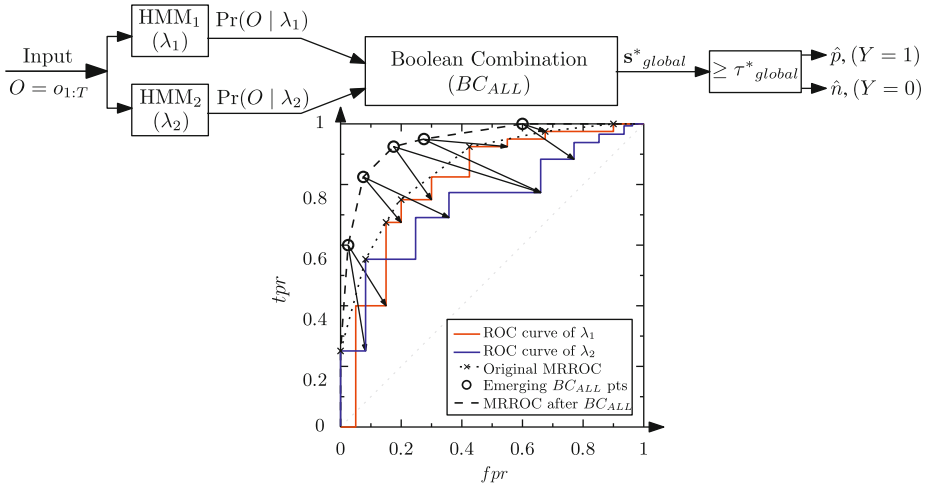

Fig. 4. Block diagram of the system used for combining the responses of two HMMs. It illustrates the combination of HMM responses in the ROC space according to the BC_{ALL} technique.

each of the 10 Boolean functions (refer to Appendix A), BC_{ALL} fuses the responses of each threshold from the first curve (R_{a_i}) with the responses of each threshold from the second (R_{b_j}).[4] Responses of the fused thresholds are then mapped to points (*fpr*, *tpr*) in the ROC space. The thresholds of points that exceeded the original ROCCH of original curves are then stored along with their corresponding Boolean functions. The ROCCH is then updated to include the new emerging points. When the algorithm stops, the final ROCCH is the new MRROC in the Newman–Pearson sense. The outputs are the vertices of the final ROCCH, where each point is the results of two thresholds from the ROC curves fused with the corresponding Boolean function. These thresholds and Boolean functions form the elements of s^*_{global}, and are stored and applied during operations, as illustrated in Fig. 4.

Algorithm 1. $BC_{ALL}(T_a,T_b,labels)$: Boolean combination of two ROC curves.

Input: Thresholds of ROC curves, T_a and T_b, and *labels* (of validation set)
Output: ROCCH and fused responses (*Rab*) of combined curves, where each point is the result of two fused thresholds along with the corresponding Boolean function (*bf*)

1 let $m \leftarrow$ number of distinct thresholds in T_a
2 let $n \leftarrow$ number of distinct thresholds in T_b
3 Allocate F an array of size: $[2, m \times n]$ // holds temporary results of fusions
4 $BooleanFunctions \leftarrow \{a \wedge b, \neg a \wedge b, a \wedge \neg b, \neg(a \wedge b), a \vee b, \neg a \vee b, a \vee \neg b, \neg(a \vee b), a \oplus b, a \equiv b\}$
5 Compute $ROCCH_{old}$ of the original curves
6 **foreach** $bf \in BooleanFunctions$ **do**
7 **for** $i = 1, \ldots, m$ **do**
8 $R_a \leftarrow (T_a \geq T_{a_i})$ // converting threshold of 1st ROC to responses
9 **for** $j = 1, \ldots, n$ **do**
10 $R_b \leftarrow (T_b \geq T_{b_j})$ // converting threshold of 2nd ROC to responses
11 $R_c \leftarrow bf(R_a, R_b)$ // combined responses with *bf*
12 Compute (*tpr*,*fpr*) using R_c and *labels*
13 Push (*tpr*,*fpr*) onto F
14
15 Compute $ROCCH_{new}$ of F
16 Store thresholds and corresponding Boolean functions that exeeded the $ROCCH_{old}$,
17 $s^*_{global} \leftarrow (T_{a_x}, T_{b_y}, bf)$ // to be used during operations
  Store the responses of these emerging points into R // to be used with BCM_{ALL} and IBC_{ALL}
  $ROCCH_{new} \leftarrow ROCCH_{old}$ // Update ROCCH
18 **Return** $ROCCH_{new}$, R, s^*_{global}

[4] For each ROC curve, the matrix of responses associated with the thresholds can be directly input to Algorithm 1 instead of converting each threshold to corresponding responses at line 8 and 10. Although it is not useful for combining two curves and it increases the memory requirements, the matrix of responses is needed to recombine resulting responses with another curve as in the following algorithms.

The BC_{ALL} technique makes no assumptions regarding the independence of the detectors. Instead of fusing the points of ROC curves under the independence assumptions (Table 1), this techniques directly fuses the responses of each decision threshold, accounting for both independent and dependent cases. In fact, by applying all Boolean functions to combine the responses for each

threshold (line 11 of Algorithm 1), it implicitly accounts for the effects of correlation (see Table 2). This is due to the direct fusion of responses, which considers the joint conditional probabilities of each detector at each threshold. Furthermore, the BC_{ALL} always provides a level of performance that is equivalent or higher than that of the MRROC of the original ROC curves. In the worst-case scenario, when the responses of detectors do not provide diverse information, or when the shape of the ROC curve on the validation set differs significantly from that of the test set, the BC_{ALL} is lower bounded by the MRROC of the original curves.

Algorithm 2. $BCM_{ALL}([T_1,\ldots,T_K],labels)$: Cumulative combination of multiple ROC curves based on BC_{ALL}.

Input: Thresholds of K ROC curves $[T_1,\ldots,T_K]$ and *labels*
Output: ROCCH of combined curves where each point is the result of the combination of combinations

1 $[ROCCH_1,R_1]=BC_{ALL}(T_1,T_2,labels)$ // combine the first two ROC curves
2 **for** $k=3,\ldots,K$ **do**
3 // combine the responses of the previous combination with those of the following ROC curve
$[ROCCH_{k-1},R_{k-1}]=BC_{ALL}(R_{k-2},T_k,labels)$
4 **Return** $ROCCH_{K-1},R_{K-1}$ and the stored tree of the selected responses/thresholds fusions along with their corresponding fusion functions

Including all Boolean functions accommodates for the concavities in the curves. Indeed, AND and OR rules will not provide improvements for the inferior points that correspond to concavities and make for an improper ROC curve, or points that are close to the diagonal line in the ROC space. Other Boolean functions, for instance those that exploit negations of responses, may however emerge. The BC_{ALL} technique can therefore be applied even when training and validation data are limited and heavily imbalanced, to combine the decisions of any soft, crisp, or hybrid detectors in the ROC space. This includes combining the responses of the same detector trained on different data or features or trained according to different parameters, or from different detectors trained on the same data, etc.

4.2. Boolean combination of multiple ROC curves

Different strategies may be implemented for combining multi-ROC curves. A commonly proposed strategy for a cumulative combination of ROC curves is to start by any pair of the ROC curves then combine the resulting responses with the third, then with the fourth and so on, until the last ROC curve [14,34]. As described in Algorithm 2, the thresholds (T_1 and T_2) of first two ROC curves are initially combined with the BC_{ALL} technique. Then, their combined responses (R_1) are directly input into line 8 of Algorithm 1 and combined with the thresholds of the third ROC curve (T_3). A pair-wise combination of ROC curves is another alternative, in which the BC_{ALL} technique is applied to each pair of ROC curves, and then the MRROC is then applied to the resulting combinations. Both strategies have been investigated and typically lead to comparable results. However, the time and memory complexity associated with the cumulative strategy can be considerably lower than for the pair-wise one. This is due to the number of permutations required in the pair-wise combinations. In additions, the pair-wise strategy requires combining all thresholds for each two curves, while combining the resulting responses with a new curve is less demanding since the number of selected responses is typically much lower than the number of thresholds. The reader is referred to Section 4.3 for additional details. The cumulative strategy for Boolean combination of multiple ROC curves (BCM_{ALL}) described in Algorithm 2 is adopted in this paper.

Further improvements in performance may be achieved by re-combining the output responses of combinations resulting from the BC_{ALL} (or BCM_{ALL}) with those of the original ROC curves over several iterations. A novel iterative Boolean combination (IBC_{ALL}) is presented in Algorithm 3 and allows for combination that maximize the AUC of K ROC curves by re-combining the previously selected thresholds and fusion functions with those of the original ROC curves. During the first iteration, the ROC curves of two or more detectors are combined using the BC_{ALL} or BCM_{ALL}. This defines a potential direction for further improvement in performance within the combination space. Then, the IBC_{ALL} proceeds in this direction by re-considering information from the original curves over several iterations. The iterative procedure accounts for potential combinations that may have been disregarded during the first iteration, and are mostly useful when provided with limited and imbalanced training data. The iterative procedure stops when there are no further improvements between the AUC of old and new ROCCHs or a maximum number of iterations are performed. This stopping criteria can be controlled by tolerating a small difference between the old and new AUC values ($\varepsilon = AUC(ROCCH_{NEW}) - AUC(ROCCH_{OLD})$), or by applying a statistical test for significance when working with several replications. Although sub-optimal, the IBC_{ALL} algorithm overcomes the impractical exponential explosion in computational complexity associated with the brute-force strategy suggested by Barreno et al. [13] (see Section 4.3).

Algorithm 3. $IBC_{ALL}([T_1,\ldots,T_K],labels)$: Iterative Boolean combination based on BC_{ALL} or BC_{ALL}

Input: Thresholds of K ROC curves $[T_1,\ldots,T_K]$ and *labels*
Output: ROCCH of combined curves where each point is the result of the combination of combinations through several iterations

1 $[ROCCH_{OLD},R_{OLD}]=BCM([T_1,T_2,\ldots,T_K],labels)$
2 **while** $(AUC(ROCCH_{NEW}) \geq AUC(ROCCH_{OLD})+\varepsilon)$ or $(numberIterations \leq maxIter)$ **do**
3 $[ROCCH_{NEW},R_{NEW}]=BC(R_{OLD},[T_1,T_2,\ldots,T_K],labels)$
4 **return** $ROCCH_{NEW},R_{NEW}$ and the stored tree of the selected responses fusions along with their corresponding fusion functions

Note that the IBC_{ALL} can also be applied to repair ROC concavities. In such scenarios, the same thresholds, say T_a, of the ROC curve to

be repaired are input twice into the IBC_{ALL} algorithm, i.e., $IBC_{ALL}(T_a,T_a,labels)$, and iterates until the AUC stops improving. After applying the IBC_{ALL} to a ROC curve the diverse information from the inferior points are taken into consideration in view improving the performance. The resulting MRROC curve is guaranteed to be proper and convex. In the worst-case scenario, the repaired is lower bounded by the ROCCH. Finally, the IBC_{ALL} repairs the concavities in a complementary way to *LCR* [19], therefore applying both techniques may yield even higher level of performance as presented in Section 6.

4.3. Time and memory complexity

Given a pair of detectors, C_a and C_b, having respectively n_a and n_b distinct thresholds on their ROC curves. During the design of the IBC_{ALL} system, the worst-case time required for fusion using BC_{ALL} is the time required for computing all 10 Boolean functions to combine those thresholds, i.e., $10 \times n_b \times n_b$. The worst-case time complexity is $\mathcal{O}(n_a n_b)$ Boolean operations. The worst-case memory requirements is an array of floating point registers of size $2 \times n_b \times n_b$ for storing the temporary results (*tpr*, *fpr*) of each Boolean function (denoted by *F* in Algorithm 1). Therefore, the worst-case memory complexity is $\mathcal{O}(n_a n_b)$.

When the number of distinct thresholds becomes very large, these thresholds can be sampled (or histogrammed) into a smaller number of bins before applying the algorithm to reduce both time and memory complexity. Nevertheless, in scenarios with limited and imbalanced data, which is the main focus of this work, the number of distinct thresholds is typically small. The BC_{ALL} is very efficient in these cases.

When the BCM_{ALL} is applied to combine the response of several ROC curves of *K* detectors, the worst-case time can be roughly stated as *K* times that of the BC_{ALL} algorithm. However, after combining the first two ROC curves, the number of emerging responses on the ROCCH, is typically very small with respect to the number of thresholds on each ROC curve. Let n_{max} be the largest number of thresholds among the *K* ROC curves to be combined. When *K* grows, it is conservative to consider the worst-case time complexity of the order of $\mathcal{O}(n_{max}^2 + K \cdot n_{max})$ Boolean operations. The worst-case time complexity for combining *K* detectors with IBC_{ALL} is that of the BCM_{ALL} multiplied with the number of iterations (*I*), $\mathcal{O}(I(n_{max}^2 + K.n_{max}))$. The worst-case memory complexity for both BCM_{ALL} and IBC_{ALL} is $\mathcal{O}(n_{max} n_{max})$. This is limited to the memory required for combining the first two ROC curves with the BC_{ALL} algorithm.

As a comparative example, consider two soft detectors C_a and C_b with their ROC curves having respectively a *small* number of distinct thresholds $n_a = 100$ and $n_b = 50$. Since each threshold on the ROC curve of a soft detector is a crisp detector, the total number of crisp detectors is therefore $n = n_a + n_b = 150$. The brute-force search for optimal combination of crisp detector is 2^{2^n} and can be reduced to 2^n as proposed by Barreno et al. [13]. The exhaustive optimal search requires a prohibitively large number, $2^{150} \approx 1.4 \times 10^{45}$, of Boolean computations.[5] This is compared to 5000 Boolean computations with BC_{ALL} and $I \times 10{,}200 \approx 10^6$ Boolean computations with IBC_{ALL}, where the number of iterations *I* is typically less than 10. Although the IBC_{ALL} algorithm is efficient, its time and memory complexity can be always reduced using the sampling technique, on the account of a small loss in the combination performance.

In contrast, during operations the system will be using one vertex or interpolating between two vertices on the final convex hull provided by IBC_{ALL} according to a specific false alarm rate. Each vertex has its own set of Boolean combination functions, which may be derived from all (or a subset of) *K* detectors that have been considered during the design phase. In practice, the computational overhead of these Boolean functions is lower than that of operating the required number of detectors. Therefore, the worst-case time and memory complexity is limited to operating the *K* detectors. When there are design constraints on the number of operational detectors *K*, they must be considered during the design phase. A larger set of detectors $\mathcal{K} > K$ could be first employed to realize an upper bound for analyzing the system performance. Then, the best subset of *K* detectors that limits the decrease of performance with reference to the upper bound can be selected for operations. This trade-off between the number of operational detectors and required performance can be a time consuming task. However since it is conducted during the design phase, various subsets selection and parallel processing techniques can be employed for a more efficient search.

4.4. Related work on classifiers combinations

Classifiers can be combined at various levels and categorized according to pre- and post-classification levels [35,36]. In pre-classification combination occurs at the sensor (or raw data) and feature levels, while post-classification combination occurs at the score, rank and decision levels. Pre-classification fusion methods are based on the generation of ensemble of classifiers (EoC), each trained on different data sets or subsets obtained by using techniques such as data-splitting, cross-validation, bagging [37], and boosting [38]. This can be also obtained by constructing classifiers that are trained on different feature subsets, for instance ensemble generation methods such as the random subspace method [39]. Static ensemble selection attempts to select the "best" classifiers from the pool based on various diversity measures [40,41], prior to combining their results. An alternative approach consists in combining the outputs of classifiers and then selecting the best performing ensemble evaluated on an independent validation set [42,43]. The later approach has proven to be more reliable than the diversity-based approach [40,43]. However, since combination is performed before selection its success depends on the chosen method(s) of combination, which may be sub-optimal.

In post-classification phase, whether the EoC is inherent to the problem at hand, generated or selected, classifier responses must be combined using a fusion function. Fusion at the score level is more prevalent in literature [44]. Normalization of the scores is typically required, which may not be a trivial task. Fusion functions may be static (e.g., sum, product, min, max, average, majority vote, etc.), adaptive (e.g., weighted average, weighted vote, etc.) or trainable (also known as stacked or meta-classifier), where another classifier is trained on classifier responses and then used as combiner [45,46]. This trainable approach may introduce overfitting and requires an independent validation set for tuning the combiner parameters. Fusion at the rank level is mostly suitable for multi-class classification problems, where the correct class is expected to appear in the top of the ranked list. Logistic regression and Borda count [47,48] are among the fusion methods at this level. Rank-level methods simplify the combiner design since normalization is not required.

Fusion at the decision level exploits the least amount of information since only class labels are input to the combiner. Compared to the other fusion methods, it is less investigated in literature. The simple majority voting rule [49] and behavior-knowledge-space (BKS) [50] are the two most representative decision-level fusing methods. One issue that appears with decision level fusion is the possibility of ties. The number of

[5] The authors stated that for only $n = 40$ detectors the computational time would require about a year and a half [13].

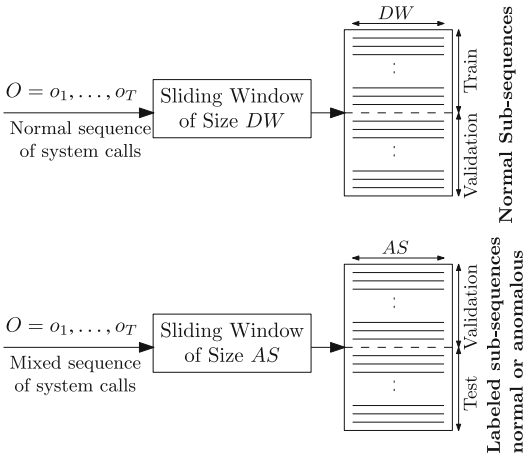

Fig. 5. Illustration of data pre-processing for training, validation and testing.

classifiers must therefore be greater than the number of classes. BKS can be only applied to low dimensional problems. Moreover, in order to have an accurate probability estimation, it requires a large number of training samples and another independent database to design the combination rules.

The proposed *IBC* in this paper provides an efficient technique which exploit all Boolean combinations as well as the MRROC interpolation for an improved performance. Combination of responses within the ROC space does not require neither re-training of dichotomizers nor normalization of scores. This is because ROC curves are invariant to monotonic transformation of classification thresholds [12]. These advantages allow the *IBC* technique to be directly applied at either the score or the decision levels.

Bayesian learning approaches have been proposed to estimate HMM parameters by integrating over the parameters rather than optimizing. For instance, variational Bayesian learning has been proposed with a suitable prior for all variables to estimate an ensemble of HMMs with the same order, each trained on a different subset of the data, to approximate the entire posterior probability distribution [51]. Nonparametric Bayesian learning have also been proposed for estimating HMMs order and other parameters from the training data [7]. Starting with a large order, the HMM parameters and the number of states are integrated out with reference to their posterior probabilities. As the method converges to a solution, redundant states are eliminated which yields to automatic order selection. These can been considered as pre-classification combinations of HMMs. HMMs with the same orders are trained and combined using different subsets of the data or HMMs with the different orders are trained and combined on the same data. Comparing the performance of these techniques to that of the *IBC* would be an interesting future work. Note however that the proposed *IBC* is a general post-classification combination technique at the response level. It can be used to combine the results of these Bayesian approaches, as well as the results of any other technique, for improved performance.

5. Experimental methodology

The experiments are conducted on both synthetically generated data and sendmail data from the University of New Mexico (UNM) data sets.[6]

[6] http://www.cs.unm.edu/~immsec/systemcalls.htm

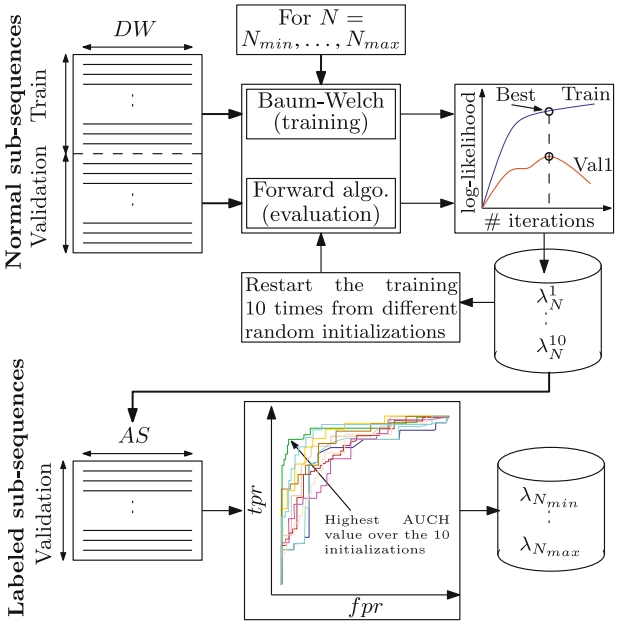

Fig. 6. Illustration of the steps involved for estimating HMM parameters.

5.1. University of New Mexico (UNM) data

The UNM data sets are commonly used for benchmarking anomaly detections based on system calls sequences [3]. In related work, intrusive sequences are usually labeled by comparing normal sequences, using STIDE matching technique. This labeling process considers STIDE responses as the ground truth, and leads to a biased evaluation and comparison of techniques, which depends on both training data size and detector window size. To confirm the results on system calls data from real processes, the same labeling strategy is used in this work. However fewer sequences are used to train the HMMs and STIDE to alleviate the bias. Therefore, STIDE is first trained on all the available normal data according to different window sizes, and then used to label the corresponding sub-sequences from the 10 sequences available for testing. The resulting labeled sub-sequences of the same size are concatenated, then divided into blocks of equal sizes, one for validation and the other for testing. During the experiments, smaller blocks of normal data (100–1000 symbols) are used for training the HMMs and STIDE. In addition to the labeling issue, the normal sendmail data are very redundant and anomalous sub-sequences in the testing data are very limited. Nevertheless, due to the limited publicly available system call data, sendmail data are the mostly used in literature.

5.2. Synthetic data

The need to overcome issues encountered when using real-world data for anomaly-based HIDS (incomplete data for training and labeling) has lead to the implementation of a synthetic data generation platform for proof-of-concept simulations. It is intended to provide normal data for training and labeled data (normal and anomalous) for testing. This is done by simulating different processes with various complexities then injecting anomalies in known locations. The data generator is based on the conditional relative entropy (*CRE*) of a source; it is closely related to the work of Tan and Maxion [52]. The *CRE* is defined as the conditional entropy divided by the maximum entropy (*MaxEnt*) of that source, which gives an irregularity index to the generated data. For two random variables x and y the CRE is given

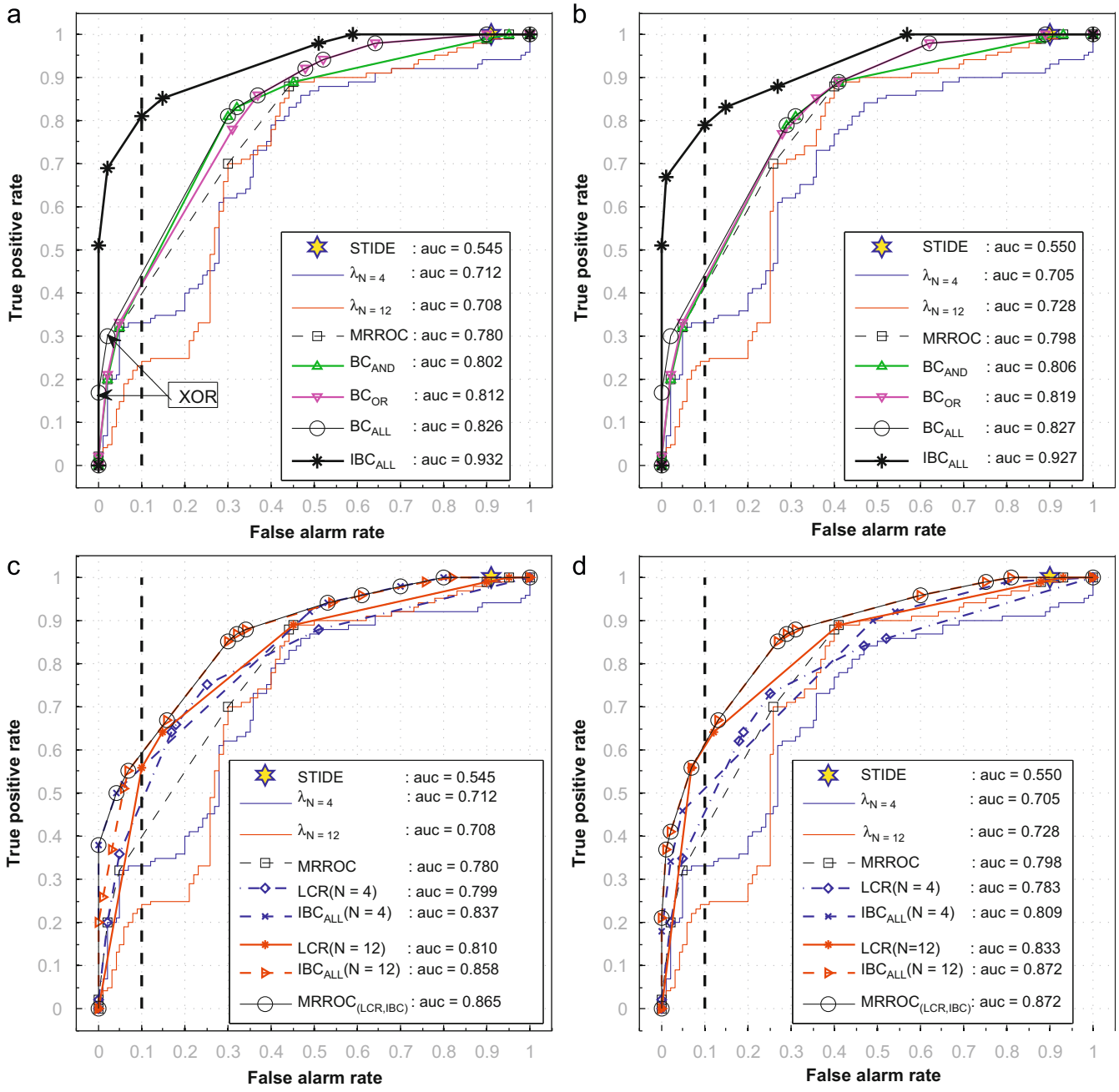

Fig. 7. Illustrative example that compares the AUC performance of techniques for combination (top) and for repairing (bottom) of ROC curves. The example is conducted on a system consisting of two ergodic HMMs trained with $N=4$ and 12 states, on a block of 100 sequences, each of length $DW=4$ symbols, synthetically generated with $\Sigma=8$ and $CRE=0.3$. (a) Combinations using the validation set, (b) combinations using the test set (c) repairing using the validation set (d) repairing using the test set.

by $CRE=-\sum_x p(x)\sum_y p(y|x)\log p(y|x)/MaxEnt$, where for an alphabet of size Σ symbols, $MaxEnt=-\Sigma\log(1/\Sigma)$ is the entropy of a theoretical source in which all symbols are equiprobable. It normalizes the conditional entropy values between $CRE=0$ (perfect regularity) and $CRE=1$ (complete irregularity or random). In a sequence of system calls, the conditional probability, $p(y|x)$, represents the probability of the next system call given the current one. It can be represented as the columns and rows (respectively) of a Markov model with the transition matrix $M=\{a_{ij}\}$, where $a_{ij}=p(S_{t+1}=j|S_t=i)$ is the transition probability from state i at time t to state j at time $t+1$. Accordingly, for a specific alphabet size Σ and CRE value, a Markov chain is first constructed, then used as a generative model for normal data. This Markov chain is also used for labeling injected anomalies as described below. Let an anomalous event be defined as a surprising event which does not belong to the process normal pattern. This type of event may be a *foreign-symbol* anomaly sequence that contains symbols not included in the process normal alphabet, a *foreign n-gram* anomaly sequence that contains *n-grams* not present in the process normal data, or a *rare n-gram* anomaly sequence that contains *n-grams* that are infrequent in the process normal data and occurs in burst during the test.[7]

Generating training data consists of constructing Markov transition matrices for an alphabet of size Σ symbols with the desired irregularity index (CRE) for the normal sequences. The

[7] This is in contrast with other work which consider rare event as anomalies. Rare events are normal, however they may be suspicious if they occurs in high frequency over a short period of time.

Table 4
Worst-case time and memory complexity for the illustrative example.

Number of iterations	1	2	3	4	5	6	7
Number of emerging points	9	10	10	11	10	9	6
Time complexity	400,000	36,000	40,000	40,000	44,000	40,000	36,000
Memory complexity	80,000	3600	4000	4000	4400	4000	3600

normal data sequence with the desired length is then produced with the Markov chain, and segmented using a sliding window (shift one) of a fixed size, *DW*. To produce the anomalous data, a random sequence ($CRE=1$) is generated, using the same alphabet size Σ, and segmented into sub-sequences of a desired length using a sliding window with a fixed size of *AS*. Then, the original generative Markov chain is used to compute the likelihood of each sub-sequence. If the likelihood is lower than a threshold it is labeled as anomaly. The threshold is set to $(\min(a_{ij}))^{AS-1}, \forall_{i,j}$, the minimal value in the Markov transition matrix to the power $(AS-1)$, which is the number of symbol transitions in the sequence of size *AS*. This ensures that the anomalous sequences of size *AS* are not associated with the process normal behavior, and hence foreign *n-gram* anomalies are collected. The trivial case of foreign-symbol anomaly is disregarded since it is easy to be detected. Rare *n-gram* anomalies are not considered since we seek to investigate the performance at the detection level, and such kind of anomalies are accounted for at a higher level by computing the frequency of rare events over a local region. Finally, to create the testing data another normal sequence is generated, segmented and labeled as normal. The collected anomalies of the same length are then injected into this sequence at random according to a mixing ratio.

Fig. 5 illustrates the data pre-processing for training, validating and testing using the UNM sendmail data or the generated data. The only difference is the ground truth for labeling, which is all the available normal data for sendmail and the generator itself for the synthetic data.

The experiments conducted in this paper using the data generator simulate a small process and a more complex process, with $\Sigma=\{8,50\}$ symbols, and $CRE=\{0.3,0.4\}$, respectively. The sizes of injected anomalies are assumed equal to the detector window sizes $AS=DW=\{2,4,6\}$. For both scenarios, the presented results are for validation and test sets that comprise 75% of normal and 25% of anomalous data.

5.3. Experimental protocol

Fig. 6 illustrates the steps involved for estimating HMM parameters. For each detector window set of size *DW*, different discrete-time ergodic HMMs are trained with various number of hidden states $N=[N_{min},\ldots,N_{max}]$. The number of symbols is taken equal to the process alphabet size. The iterative Baum–Welch algorithm is used to estimate HMM parameters [22] using the training data, which only comprises normal sequences. To reduce overfitting effects, the evaluation of the log-likelihood, using the Forward algorithm [22], on an independent validation set also comprising only normal sequences is used as a stopping criterion. The training process is repeated 10 times using a different random initialization to avoid local minima. The log-likelihood of a second validation set comprising normal and anomalous sequences is then evaluated by each HMM, which provides 10 ROC curves. Finally, the model that gives the highest area under its convex hull is selected, which results an HMM for each *N* value. When working with the synthetic data, this procedure is replicated 10 times with different training, validation and testing sets, and the results are averaged and presented along with the standard deviations to provide a statistical confidence intervals.

The performance obtained by fusion of μ-HMMs in ROC space with the proposed BC_{ALL} technique is compared to that of MRROC fusion of the original models, and to that of the conjunction (BC_{AND}) and disjunction (BC_{OR}) combinations [14,16]. This is also compared to the performance of the IBC_{ALL} technique applied to combine the μ- HMMs through several iterations. In addition, the performance of STIDE is shown as reference. To investigate the effect of repairing the concavities in ROC curves, each ROC curve is first repaired using the IBC_{ALL} and *LCR* [19] techniques, and then all curves are combined with the MRROC technique.

The area under the ROC curve (AUC) has been proposed as more robust scalar summary of classifiers performance than accuracy [26,53,54]. The AUC assesses the ranking in terms of class separation, i.e., evaluates how well a classifier is able to sort its predictions according to the confidence it assigns to them. For instance, with an $AUC=1$ all positives are ranked higher than negatives indicating a perfect discrimination between classes. A random classifier has an $AUC=0.5$ that is both classes are ranked at random. For a crisp classifier, the AUC is simply the area under the trapezoid and is calculated as the average of the *tpr* and *fpr* values. For a soft classifier, the AUC may be estimated directly from the data either by summing up the areas of the underlying trapezoids [12] or by means of the Wilcoxon–Mann–Whitney (WMW) statistic [20]. When the ROC curves cross, it is possible for a high-AUC classifier to perform *worse* in a specific region of ROC space than a low-AUC classifier. In such case, the partial area under the ROC curve (pAUC) [21] could be useful for comparing the specific regions of interest [55]. If the AUCs (or the pAUCs) are not significantly different, the shape of the curves might need to be looked at. It may also be useful to look at the *tpr* for a fixed *fpr* of particular interest. Since the MRROC can be applied to any ROC curve, the performance measures in all experiments are taken with reference to the ROCCH. This includes the area under the convex hull (AUCH), the partial area under the convex hull for the range of $fpr=[0, 0.1]$ ($AUCH_{0.1}$), and the *tpr* at a fixed $fpr=0.1$.

6. Simulation results and discussion

6.1. An illustrative example with synthetic data

Fig. 7 presents an example of the impact on performance obtained after applying different techniques for combining and repairing ROC curves: MRROC, *BC* and *IBC*. The training, validation and testing data are generated synthetically as described in Section 5.2, with an alphabet of size $\Sigma=8$ symbols and with a $CRE=0.3$. The training and validation of the ergodic HMMs is carried out according to the methodology described in Section 5. A block of data of size 100 sequences, each of size $DW=4$, is used to train HMMs, and another validation block of the same size is used to implement a stopping criterion. Each validation and test set is composed of 200 sequences, each of size $AS=4$. In both cases, the ratio of normal to anomalous sequences is four to one.

For improved visibility, Fig. 7 only shows the combinations of two HMMs, each one trained with different number of states, $N=4$ and 12. The ROC curves for the two HMMs along with their MRROC are presented for the validation (Fig. 7(a)) and test (Fig. 7(b)) data sets. The performance of STIDE is also shown for reference. To visualize

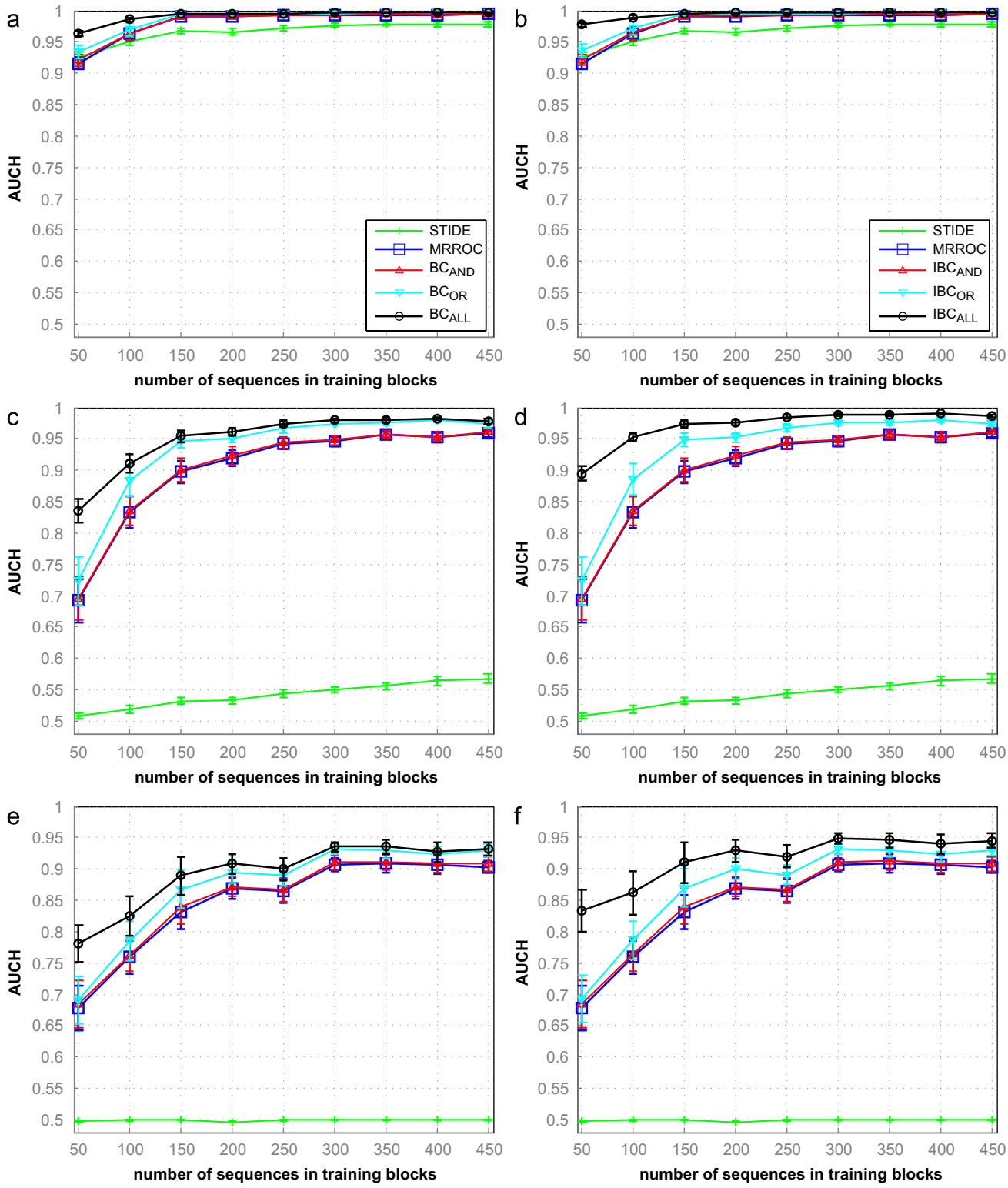

Fig. 8. Results for synthetically generated data with $\Sigma = 8$ and $CRE = 0.3$. Average AUC values obtained on the test sets as a function of the number of training blocks for a μ-HMM where the three HMMs are trained with a different state ($N = 4,8,12$), and combined with the MRROC, *BC* and *IBC* techniques. Average AUC performance is compared for various training block sizes (50–450 sequences) and detector windows sizes ($DW = 2,4,6$). Error bars are standard deviations over 10 replications. (a) BC ($DW = 2$), (b) IBC ($DW = 2$), (c) BC ($DW = 4$), (d) IBC ($DW = 4$), (e) BC ($DW = 6$) and (f) IBC ($DW = 6$).

the impact on performance of combining with AND and OR Boolean functions separately [14,16], Figs. 7(a) and (b) show the results with BC_{AND} and BC_{OR} along with BC_{ALL} and IBC_{ALL}. In Figs. 7(c) and (d), IBC_{ALL}, and *LCR* [19] are applied to repair each ROC curve, and the resulting curves are then combined with the MRROC technique.

As expected, STIDE performs poorly since the data provided for training is limited and the database memorized by STIDE only comprises a fraction of the normal behavior. This contrasts with the generalization capabilities of HMMs. As shown in Figs. 7(a) or (b), the improvement in AUC performance achieved by applying

BC_{AND} and BC_{OR} is modest. This is also seen when comparing their *tpr* values at an operating point of, for instance, $fpr=0.1$ with respect to the MRROC of the original ROC curves. Indeed, these AND and OR Boolean functions are unable to exploit information in inferior points. In this case, the BC_{ALL} does not provide a considerably higher level of performance than the BC_{AND} and BC_{OR}, this is not typically the case, but was deliberately selected for this example. However, as shown in Fig. 7(a), two additional combination rules have emerged from the *XOR* Boolean function with BC_{ALL}. Diverse information from the emerging combination rules (resulting, in this case, from *AND*, *OR* and *XOR* Boolean functions) serve as a guide, within the combination space, to a further performance improvement with the IBC_{ALL} technique. Indeed, after seven iterations, IBC_{ALL} is able to achieve a considerably higher level of performance by recombining emerging rules from each iteration with the original curves. IBC_{ALL} iterative procedure repeatedly selects diverse information and accounts for potential combinations that may have been disregarded during previous iterations.

In this example, the validation set is composed of 200 sequences and the ROC curves of both HMMs ($\lambda_{N=4}$ and $\lambda_{N=12}$) contain the same number of unique thresholds,[8] 200. The worst-case time complexity involved with BC_{ALL} is the time required to compute all 10 Boolean functions for each combination of thresholds, i.e., $10\times200\times200=400{,}000$ Boolean operations. The worst-case memory complexity is an array of floating point registers of size $2\times200\times200=80{,}000$, holding the temporary results (*tpr*, *fpr*) of each Boolean function. This consists the first iteration of IBC_{ALL} and results in nine emerging combinations (or points on the ROCCH) as shown in Fig. 7(a). The time complexity of the second iteration is reduced to the time required for computing $10\times9\times(200+200)=36{,}000$ Boolean operations, and the memory complexity is reduced to $2\times1800=3600$ floating point registers. As shown in Table 4, the time and memory complexity of successive iterations are reduced by order of magnitude compared to the first iteration. However, as shown in Fig. 7(a), the level of performance is significantly improved due to these low cost iterations. During operations, the number of Boolean combinations varies with the selected operational point, however it is upper-bounded by the number of iterations. For instance, in this example a maximum of seven Boolean functions must be applied to HMMs response to achieve the desired performance as in Fig. 7(b). The time complexity of these Boolean combinations is negligible compared to that of computing the log-likelihood of the test sequences with HMM.

When using the IBC_{ALL} technique to repair the ROC curve belonging to the HMM trained with $N=12$ states, the resulting curve $IBC_{ALL}(N=12)$ dominates other ROC curves on the test set. This is because its test set performance exceeds its expected validation set performance, while the expected $IBC_{ALL}(N=4)$ performance decreases on the validation set. Similarly, the expected validation performance of the *LCR* technique can be largely different from that of the test set as shown for $LCR(N=12)$ curve. In general, the performance of both repairing techniques is less robust to variances between validation and test sets, since repairing relies on the shape of one ROC curve. In contrast, combining several ROC curves according to IBC_{ALL} technique is more robust to such variance since the interactions among all ROC curves are considered.

A closer look at ROC-based repairing techniques shows that IBC_{ALL} and *LCR* are complimentary. For instance, ROC curves repaired for $\lambda_{N=4}$, $IBC_{ALL}(N=4)$ and $LCR(N=4)$, cross in both validation and test sets. In general, IBC_{ALL} performs well when the concavities are located in the bottom-left or in the top-right corners of the ROC space. IBC_{ALL} exploits the asymmetry in the ROC curve shape caused by an imbalanced variances at the head or tail of class distributions. The *LCR* technique is efficient for repairing concavities that are located close to non-major diagonal, since it only considers the shape of the ROC curve when negating the responses of the largest concavity. A higher level of AUC performance can therefore be achieved by using the MRROC to combine ROC curves repaired using the IBC_{ALL} and *LCR* techniques.

The attempt to combine the responses of the ROC curves repaired with the IBC_{ALL} or *LCR* techniques using IBC_{ALL}, was not successful due to the low level of robustness of the repairing techniques. Combination are based on the few points that define repaired ROC curve points on the validation set, which may be in different locations on the test set. In contrast, the direct combination of ROC curves according to BC_{ALL} or IBC_{ALL} techniques starts with existing thresholds on the validation set. Although ROC curves thresholds may change on the test set, these algorithms proceed by using neighboring thresholds.

6.2. Results with synthetic and real data

Fig. 8 shows the average AUC performance on the test sets versus the number of training blocks of a μ- HMM system where each HMM is trained with a different number of states ($N=4,8,10$). Results are produced for synthetically generated data with $\Sigma=8$ and $CRE=0.3$, and for various training set sizes (50–450 symbols) and detector window sizes ($DW=2,4,6$). The performance of the composite system obtained with the MRROC combination for the original HMM ROC curves, is compared to those obtained with the BC_{AND}, BC_{OR}, BC_{ALL}[9] and IBC_{ALL} techniques. The AUC performance of STIDE is also shown for reference. The reader is referred to Appendix B for additional supporting results—$AUCH_{0.1}$ and *tpr* values at $fpr=0.1$.

As shown in these figures, the BC_{ALL} can significantly improve the AUC over the MRROC in all the presented cases. The performance of the MRROC approaches that of BC_{ALL} technique when the training data become abundant for the problem at hand, or when test cases are easily detected (e.g., when classes are well separated). For example, this is shown in Fig. 8(a) when the training block size grows beyond 150 sequences, even STIDE is able to achieve a high level of performance on this simple scenario (that is rarely encountered in real-world applications).

Although the BC_{AND} and BC_{OR} fusion were able to increase the performance over the MRROC, their performances is often significantly lower than that of the BC_{ALL}, as shown in Figs. 8(c) and (d). The significant increase in performance achieved with the BC_{OR} supports prior conclusion by Tao and Veldhuis [14]. The authors recommend using the *OR* Boolean fusion for detecting outliers in biometrics applications. However, this may not hold true when the number of positive samples is very limited.

The IBC_{ALL} technique provides the highest level of performance over all the range of conducted experiments. Since it includes the BC_{ALL} in its first iteration, the performance of IBC_{ALL} is lower bounded by that of the BC_{ALL}. Results indicate that IBC_{ALL} is most suitable for cases in which data are limited and imbalanced, as shown in the first blocks of Figs. 8(b), (d), and (f) (see additional results in Appendix B).

The performance of IBC_{ALL} improves significantly over that of the BC_{ALL}. This is due to the iterative nature of the IBC_{ALL} that is able to repeatedly exploit the information residing in the inferior points, and hence select better combinations.

[8] In many cases, the number of unique thresholds could be lower than the number of samples in the validation set.

[9] For simplicity, BC_{ALL} is also used to indicate BCM_{ALL} when combining multiple ROC curves.

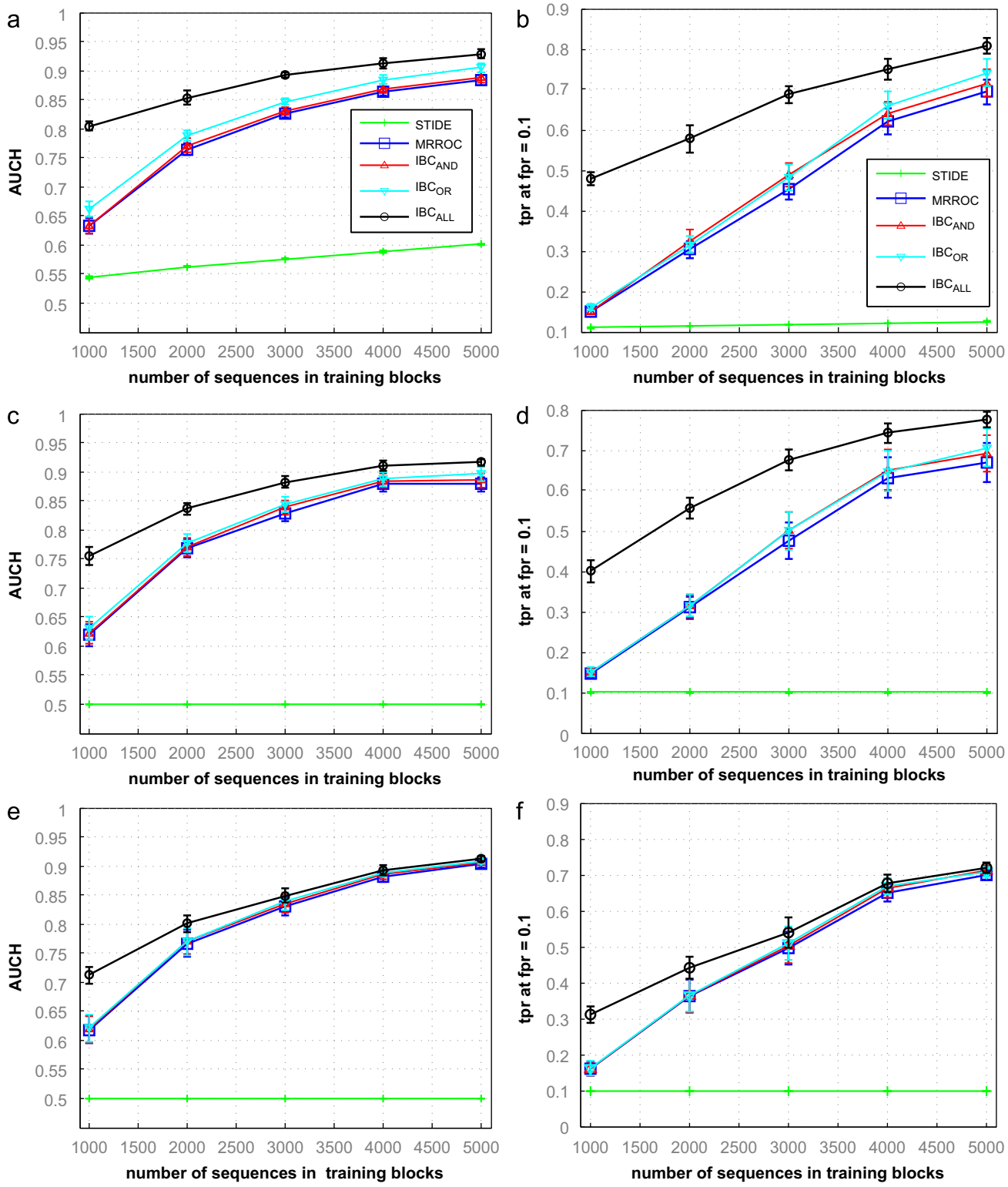

Fig. 9. Results for synthetically generated data with $\Sigma = 50$ and $CRE = 0.4$. Average AUC values (left) and *tpr* values at $fpr = 0.1$ (right) obtained on the test sets as a function of the number of training blocks for a μ-HMM where the three HMMs are trained with a different state ($N = 40,50,60$), and combined with the MRROC and *IBC* techniques. The performance is for various training block sizes (1000–5000 sequences) and detector window sizes ($DW = 2,4,6$). Error bars are standard deviations over 10 replications. (a) DW = 2, (b) DW = 2, (c) DW = 4, (d) DW = 4, (e) DW = 6 and (f) DW = 6.

When the number of blocks for training is limited, the performance of the IBC_{ALL} technique is higher than other combination techniques. In such cases, each HMM trained with a different order provides diverse information by capturing different data structure, which allows to increase the performance of IBC_{ALL}. When the amount training data become abundant, the HMMs tend to achieve the same performance with less diversity in their responses, which degrades the performance achieved by IBC_{ALL}.

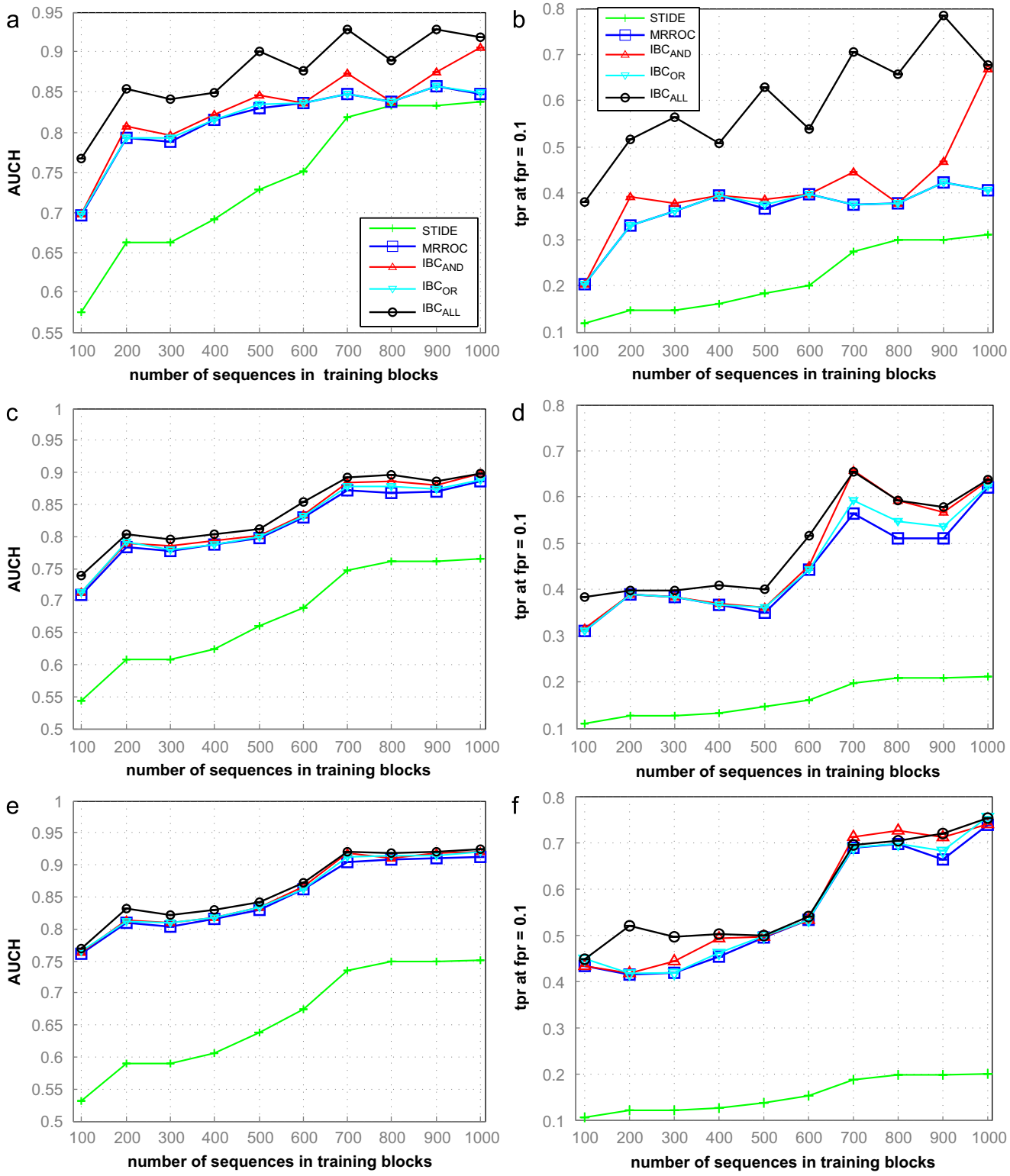

Fig. 10. Results for sendmail data. AUC values (left) and *tpr* values at $fpr=0.1$ (right) obtained on the test sets as a function of the number of training blocks for a μ-HMM where the five HMMs are trained with a different state ($N=40,45,50,55,60$), and combined with the MRROC and *IBC* techniques. The performance is compared for various training block sizes (100–1000 sequences) and detector window sizes ($DW=2,4,6$). (a) DW=2, (b) DW=2, (c) DW=4, (d) DW=4, (e) DW=6 and (f) DW=6.

With the increase of detector window size,[10] the likelihood of anomalous sequences becomes smaller at a faster rate than normal ones, and hence increases HMM detection rate [9]. As a consequence, HMMs responses become less diverse yielding to a decrease of IBC_{ALL} performance. The impact of *DW* on performance is illustrated with the second and more complex scenario below.

Since the IBC_{ALL} technique incorporates all combinations employed within the BC_{ALL} technique, only results of IBC_{ALL} are

[10] For simplicity, the detector window size, *DW*, is assumed equal to the anomaly size, *AS*.

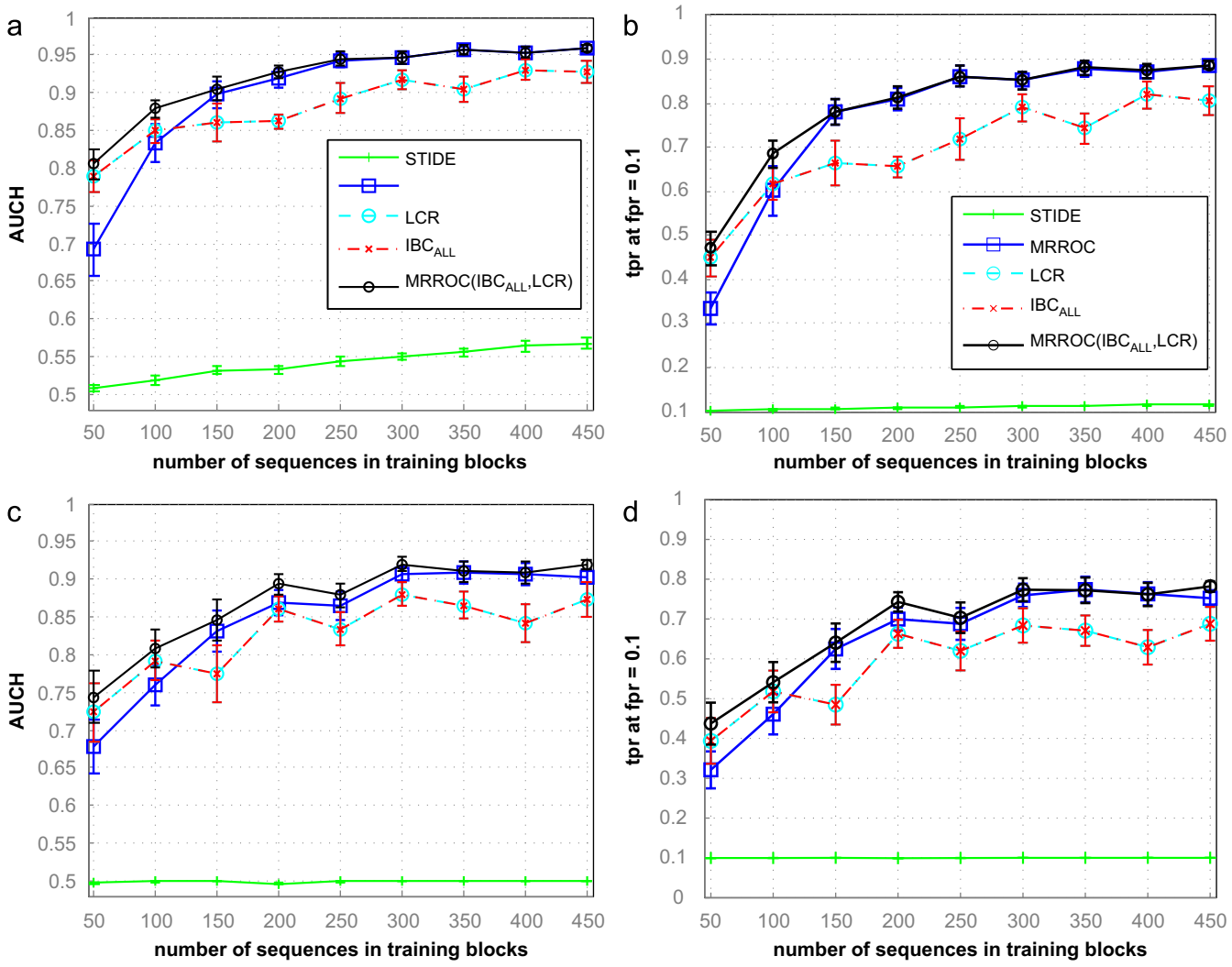

Fig. 11. Performance versus the number of training blocks achieved after repairing concavities for synthetically generated data with $\Sigma = 8$ and $CRE = 0.3$. Average AUCs (left) and *tpr* values at $fpr = 0.1$ (right) on the test set for a μ-HMM where each HMM is trained with different number of states ($N = 4,8,12$). HMMs are combined with the MRROC technique and compared to the performance of IBC_{ALL} and *LCR* repairing techniques, for various training block sizes (50–450 sequences) and detector windows sizes ($DW = 4$ and 6). (a) DW = 4, (b) DW = 4, (c) DW = 6 and (d) DW = 6.

compared to those of MRROC, IBC_{AND} and IBC_{OR} for the second synthetic scenario ($\Sigma = 50$, $CRE = 0.4$) and for sendmail data.

Fig. 9 confirms the results of Fig. 8 on the second synthetic and more complex scenario, where $\Sigma = 50$ and $CRE = 0.4$. Again the IBC_{ALL} technique provides higher level of performance than the MRROC, IBC_{AND} and IBC_{OR} techniques. In this scenario, although the results of the AND and OR Boolean combinations were allowed to iterate until convergence, their achieved performances are still significantly lower than that of the IBC_{ALL}, as shown in Fig. 9. This demonstrates the impact on performance of employing and iterating all Boolean functions until convergence. The performance gain is best illustrated for the first five training blocks (1000–3000 sequences), where the HMMs trained with different orders provide IBC_{ALL} with diverse responses for a significantly improved performance. Increasing the number of training blocks however increases the detection capabilities of the HMMs and reduces the diversity in their responses, which decreases the level of performance achieved with the IBC_{ALL} technique. As discussed previously, HMM detection ability increases with the detector window size (or anomaly size) which reduces the diversity in the μ-HMMs system. This negatively affects the performance achieved by the IBC_{ALL} technique as shown in Fig. 9.

This is also confirmed on the sendmail data in Fig. 10. Note however that with sendmail, the training data are very redundant and the test samples are very limited. Even STIDE performance was moderate with only up to 1000 training sequences, out of the available 1.5 million sequences used for labeling. Nevertheless, the difference in performance between validation and test sets, not shown for improved visibility, is significantly large where the limited test samples are split into half for testing and half for validation. Although the IBC_{ALL} is able to increase the performance, this increase is not as prominent as in the synthetic cases. This is mainly due to redundancy in the training data, where the HMMs trained with different number of states were not able to capture enough diversity in the underlying data structure. Effective combination technique must exploit diverse and complimentary information to improve systems performance.

Overall, the AUCs tend to increase to a comparable level as the number of training blocks increases. With a sufficient amount of training data, all HMMs are capable of achieving an equal level of performance, however with less diverse and complimentary

a

C_a	C_b	$\neg C_a$	$\neg C_b$	One	$Zero$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	1	0

Depend on one variable | Constant operations

De Morgan's laws:

$$\neg(C_a \wedge C_b) = \neg C_a \vee \neg C_b$$
$$\neg(C_a \vee C_b) = \neg C_a \wedge \neg C_b$$

Symbols:

$\neg = NOT$
$\wedge = AND$
$\vee = OR$
$\oplus = XOR$
$IMP = Implication$
$EQV = Equivalence$

b

		AND			$NAND$	OR	$C_a\ IMP\ C_b$	$C_b\ IMP\ C_a$	NOR	XOR	EQV
C_a	C_b	$C_a \wedge C_b$	$\neg C_a \wedge C_b$	$C_a \wedge \neg C_b$	$\neg(C_a \wedge C_b)$	$C_a \vee C_b$	$\neg C_a \vee C_b$	$C_a \vee \neg C_b$	$\neg(C_a \vee C_b)$	$C_a \oplus C_b$	$\neg(C_a \oplus C_b)$
0	0	0	0	0	1	0	1	1	1	0	1
0	1	0	1	0	1	1	1	0	0	1	0
1	0	0	0	1	1	1	0	1	0	1	0
1	1	1	0	0	0	1	1	1	0	0	1

Called minterms: they represent the four miminum classes | Called maxterms: they represent the four maximum classes | Operations with a "checkerboard" truth table

Fig. 12. All distinct Boolean operations on two variables.

information. On the other hand, the IBC_{ALL} performance will outperform others for small training set and detector window sizes. When the number of blocks for training is limited, the performance of the IBC_{ALL} technique is higher than other combination techniques. In such scenarios, each HMM trained with a different number of states is able to capture different underlying structures of data. These HMMs provide diverse information, which allows to increase the performance of IBC_{ALL}. This holds true for different detector window sizes (DW), although the impact of combinations on performance degrades with the increase of DW values. HMMs are more capable of detecting larger anomalies, providing therefore less diverse responses to the IBC_{ALL} technique.

Increasing the number of HMMs trained with different orders (N) in the μ-HMM system has a significant impact on the performance achieved with the IBC_{ALL} technique due to the added diversity among the combined HMMs. Since diversity measures and the creation of diverse ensembles are not yet well defined in literature [41,56], one can simply combine the responses of HMMs trained using a wide range of states to have an upper bound on performance. This is feasible in short period of time due to the efficiency of IBC_{ALL}. During operations however, simplified combination rules and fewer HMMs may be favored for speed constraints on the detection system. In such cases, the number of HMMs involved in the μ-HMM system can be reduced by selecting a subset of N values that does not significantly affect the upper bound achieved on performance (as described in Section 4.3). Training different HMMs on different subsets of the data provides other sources of diversity. Future work involves combining the responses of HMMs trained with different orders on different subsets of the data according to the IBC_{ALL} technique.

Finally, Fig. 11 shows results for repairing concavities using the first synthetically generated scenario ($\Sigma = 8$, $CRE = 0.3$). The μ-HMM system is trained with three different states ($N = 4,8,12$), for various training set sizes and detector window sizes, and combined with the MRROC technique. Then, the IBC_{ALL} and LCR techniques are applied to repair the concavities presented in each ROC curve associated with an HMM. The repaired ROC curves are then combined according to the MRROC. In addition, the results of IBC_{ALL} and LCR techniques are also combined with the MRROC.

Fig. 11 shows that each of the repairing techniques, IBC_{ALL} and LCR, is able to exceed the MRROC of the original curves at the first blocks. This is because in such cases the ROC curves comprise large concavities due to the limited training data. However, when the amount of training data increases, IBC_{ALL} and LCR are not able to provide a higher performance than the MRROC of the original curves unless their responses are themselves combined with the MRROC technique. This noticeable increase in performance, achieved by combining the repaired curves according to IBC_{ALL} and LCR with the MRROC, clearly indicates the complementarity of both techniques as discussed in Section 6.1.

However, the performance achieved by combining the repaired curves according to IBC_{ALL} and LCR with the MRROC is still lower than that of combining the original ROC curves using IBC_{ALL}, as presented in Figs. 8(d) and (f). Nevertheless, in results not shown in this paper, repairing with the IBC_{ALL} and LCR techniques have shown lack of robustness to changes between the validation and test sets (as discussed in Section 6.1). In addition, repairing relies on large ROC concavities and they are not designed to improve the performance when the ROC curve comprises small concavities, but represent a poor performance (e.g., parallel to the diagonal of chance). In contrast, combining ROC curves using the IBC_{ALL} may allow to exploit this information to increase performance. In practice, when a ROC curve of a detector presents concavities, performance could be improved by combining the responses of both repairing techniques, IBC_{ALL} and LCR, using the MRROC fusion. Otherwise, the IBC_{ALL} could be directly applied to combine the responses of several ROC curves and provide a higher level of performance than other techniques.

7. Conclusions

This paper presents an iterative Boolean combination (IBC) technique for efficient fusion of the responses from multiple classifiers in the ROC space. The IBC efficiently exploits *all* Boolean functions applied to the ROC curves and requires no prior assumptions about conditional independence of detectors or convexity of ROC curves. Although it seeks a sub-optimal set of combinations, the IBC is very efficient in practice and it provides a higher level of performance than related ROC-based combination techniques, especially when training data is limited and test data is heavily imbalanced. Its time complexity is linear with the number of classifiers while, the memory requirement is independent of the number of classifier, which allows for a large number of combinations. The proposed IBC is general in that it can be

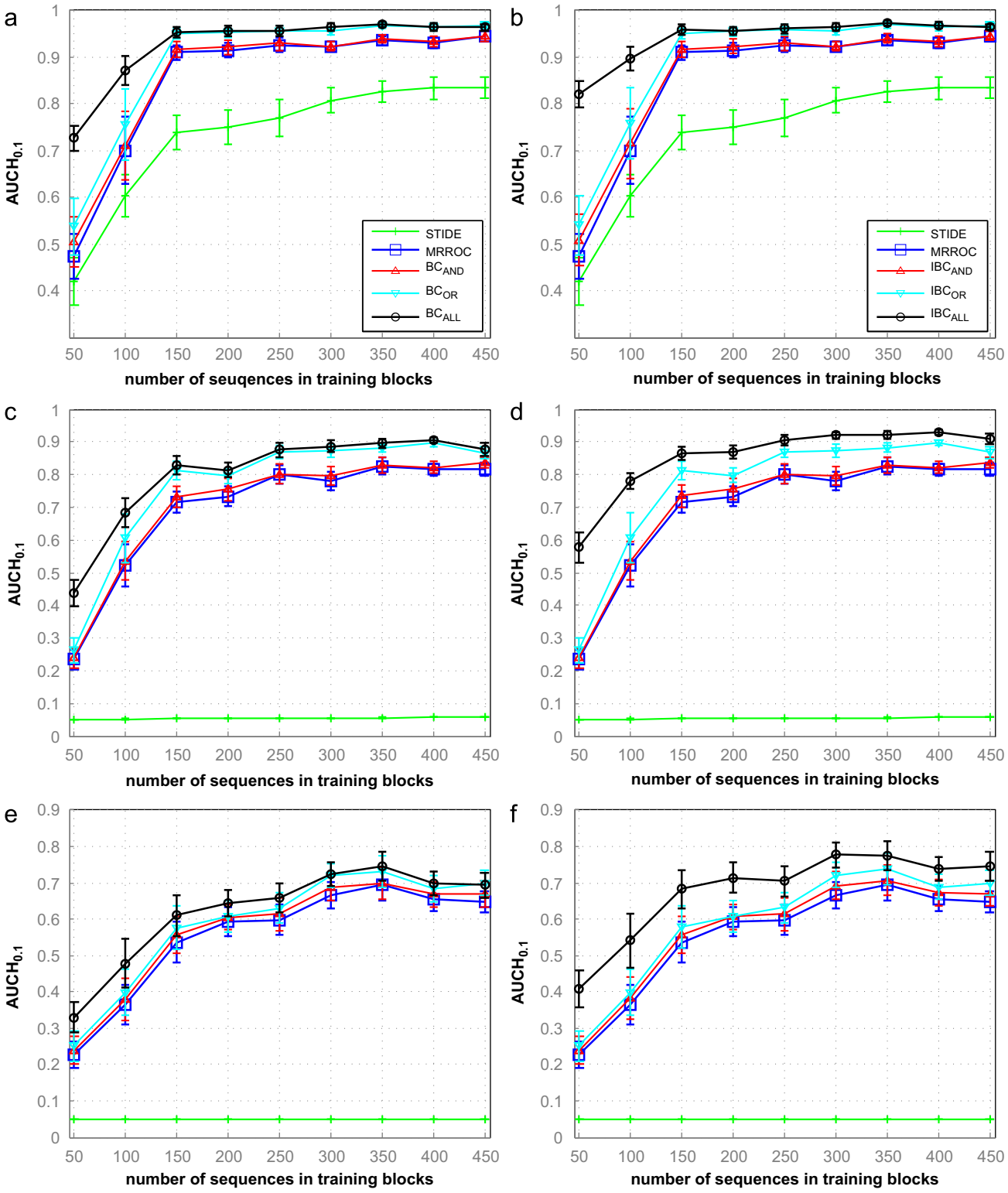

Fig. 13. Results for synthetically generated data with $\Sigma = 8$ and $CRE = 0.3$. Average $AUCH_{0.1}$ values obtained on the test sets as a function of the number of training blocks for a μ-HMM where the three HMMs are trained with a different state ($N = 4,8,12$), and combined with the MRROC, *BC* and *IBC* techniques. Average $AUCH_{0.1}$ performance is compared for various training block sizes (50–450 sequences) and detector window sizes ($DW = 2,4,6$). Error bars are standard deviations over 10 replications. (a) BC, (b) IBC, (c) BC, (d) IBC, (e) BC and (f) IBC.

employed to combine diverse responses of any crisp or soft one- or two-class classifiers, within a wide range of application domains. This includes combining the responses of the same classifier trained on different data or features or trained according to different parameters, or from different classifiers trained on the same data, etc. It is also useful for repairing the concavities in a ROC curve.

During simulations conducted on both synthetic and real HIDS data sets, the *IBC* has been applied to combine the responses a of multiple-HMM system, where each HMM is trained using a

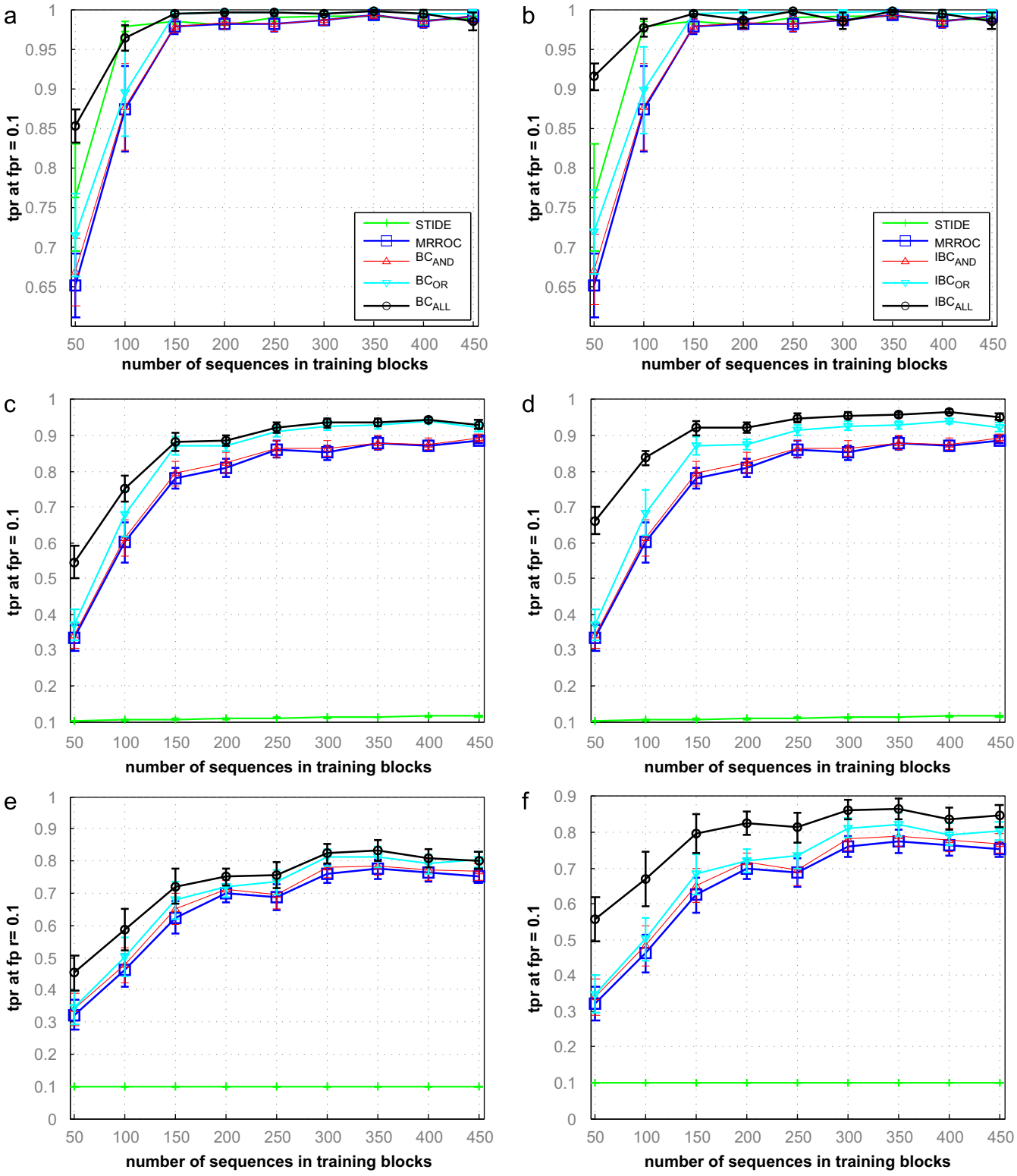

Fig. 14. Results for synthetically generated data with $\Sigma = 8$ and $CRE = 0.3$. Average *tpr* values at $fpr = 0.1$ values obtained on the test sets as a function of the number of training blocks for a μ-HMM where the three HMMs are trained with a different state ($N = 4,8,12$), and combined with the MRROC, *BC* and *IBC* techniques. Average performance is compared for various training block sizes (50–450 sequences) and detector window sizes ($DW = 2,4,6$). Error bars are standard deviations over 10 replications. (a) BC, (b) IBC, (c) BC, (d) IBC, (e) BC and (f) IBC.

different order (number of states), and capturing different temporal structures of the data. Results indicate that the *IBC* significantly improves the overall system performance over a wide range of training set sizes with various alphabet sizes and complexities of monitored processes, and according to different anomaly sizes, without a significant computational and storage overhead. Results have shown that, even with one iteration, the *IBC* technique always increases system performance over the MRROC fusion, and over the Boolean conjunction and disjunction combinations. When the *IBC* is allowed to iterate until

convergence, the system performance improves significantly and the time and memory complexity required for each iteration are reduced by an order of magnitude with reference to the first iteration. The performance gain, especially when provided with limited training data, is due to the ability of the *IBC* technique to exploit diverse information residing in inferior points on the ROC curves, which are disregarded by the other techniques. In addition, repairing the concavities in a ROC curve using the *IBC* technique can yield higher level of performance than with the MRROC technique alone. The impact on performance of repairing the concavities is shown to be comparable and complementary to an existing repairing technique that relies on inverting the largest concavity section. Therefore, combining the results of both repairing techniques according to the MRROC fusion yields a higher level of performance than applying each repairing technique alone.

In this paper, the proposed *IBC* technique is applied to combine responses from the same one-class classifiers (HMMs) trained on the same data and using the same features, however according to different orders. Future work involves applying the *IBC* to other real-world applications where the same classifiers are trained using different subsets of data and features and also to combine the responses from different classifiers trained on the same data. Investigation of diversity measures and impact of correlations within the ROC space is an important future direction. This may yield further insight into the design of classifier ensembles that contribute toward efficient and accurate combinations.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Le Fonds québécois de la recherche sur la nature et les technologies.

Appendix A. Boolean functions

Fig. 12 presents all the distinct Boolean functions on two variables. Fig. 12(a) shows the four Boolean operations that depend on one variable (the inputs and their negations) along with the two constant operations (always positive and always negatives). Fig. 12(b) represents the 10 operations that take on two variables. The first four are obtained from conjunction with some subset of its inputs negated. Similarly for the next four which are obtained from disjunction however. The last two (the XOR and its negation, EQV) are obtained with a "checkerboard" truth table. In general, for two inputs variables ($n=2$) with two possible binary outputs, there are $2^{2^n}=16$ distinct Boolean operations, whereas 10 are effectively used for combining.

Appendix B. Additional results

Figs. 13 and 14 present the average performance in terms of the partial area under the convex hull for the range of $fpr=[0, 0.1]$ ($AUCH_{0.1}$), and the tpr at a fixed $fpr=0.1$, respectively. These additional results are provided for a μ-HMM with three HMMs, each one trained with a different number of states ($N=4,8,12$), and using the synthetically generated data with $\Sigma=8$ and $CRE=0.3$. These results complement those shown in Fig. 8 (Section 6.2).

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection for discrete sequences: a survey, Technical Report TR 09-015, University of Minnesota, Department of Computer Science and Engineering, 2009.

[2] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, A sense of self for Unix processes, in: Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, 1996, pp. 120–128.

[3] C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusions using system calls: alternative data models, in: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 1999, pp. 133–45.

[4] L. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[5] B. Gao, H.-Y. Ma, Y.-H. Yang, HMMs (Hidden Markov Models) based on anomaly intrusion detection method, in: Proceedings of 2002 International Conference on Machine Learning and Cybernetics, vol. 1, 2002, pp. 381–385.

[6] X. Hoang, J. Hu, An efficient Hidden Markov Model training scheme for anomaly intrusion detection of server applications based on system calls, in: IEEE International Conference on Networks, ICON, vol. 2, Singapore, 2004, pp. 470–474.

[7] M.J. Beal, Z. Ghahramani, C.E. Rasmussen, The infinite Hidden Markov Model, in: Advances in Neural Information Processing Systems (NIPS) 2001, vol. 14, MIT Press, Cambridge, MA, 2002, pp. 577–585.

[8] J.V. Gael, Y. Saatci, Y.W. Teh, Z. Ghahramani, Beam sampling for the infinite Hidden Markov Model, in: Proceedings of the 25th International Conference on Machine Learning, ACM, Helsinki, Finland, 2008, pp. 1088–1095.

[9] W. Khreich, E. Granger, R. Sabourin, A. Miri, Combining Hidden Markov Models for anomaly detection, in: International Conference on Communications (ICC), Dresden, Germany, 2009.

[10] M.J.J. Scott, M. Niranjan, R.W. Prager, Realisable classifiers: improving operating performance on variable cost problems, in: P.H. Lewis, M.S. Nixon (Eds.), Proceedings of the Ninth British Machine Vision Conference, vol. 1, University of Southampton, UK, 1998, pp. 304–315.

[11] F. Provost, T. Fawcett, Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1997, pp. 43–48.

[12] T. Fawcett, ROC graphs: Notes and practical considerations for researchers, Technical Report HPL-2003-4, HP Laboratories, Palo Alto, CA, USA, 2004.

[13] M. Barreno, A. Cardenas, D. Tygar, Optimal ROC for a combination of classifiers, in: Advances in Neural Information Processing Systems (NIPS), vol. 20, 2008.

[14] Q. Tao, R. Veldhuis, Threshold-optimized decision-level fusion and its application to biometrics, Pattern Recognition 41 (5) (2008) 852–867.

[15] W.B. Langdon, B.F. Buxton, Evolving receiver operating characteristics for data fusion, in: EuroGP '01: Proceedings of the 4th European Conference on Genetic Programming, Springer-Verlag, London, UK, 2001, pp. 87–96.

[16] S. Haker, W.M. Wells, S.K. Warfield, I.-F. Talos, J.G. Bhagwat, D. Goldberg-Zimring, A. Mian, L. Ohno-Machado, K.H. Zou, Combining classifiers using their receiver operating characteristics and maximum likelihood estimation, in: Medical Image Computing and Computer Assisted Intervention (MICCAI), vol. 3749, 2005, pp. 506–514.

[17] J. Hill, M. Oxley, K. Bauer, Receiver operating characteristic curves and fusion of multiple classifiers, in: Proceedings of the 6th International Conference on Information Fusion, vol. 2, 2003, pp. 815–822.

[18] M. Oxley, S. Thorsen, C. Schubert, A Boolean Algebra of receiver operating characteristic curves, in: 10th International Conference on Information Fusion, 2007, pp. 1–8.

[19] P.A. Flach, S. Wu, Repairing concavities in ROC curves, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI, 2005, pp. 702–707.

[20] J. Hanley, B. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.

[21] S.D. Walter, The partial area under the summary ROC curve, Statistics in Medicine 24 (13) (2005) 2025–2040.

[22] L.E. Baum, G.S. Petrie, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, The Annals of Mathematical Statistics 41 (1) (1970) 164–171.

[23] Y. Ephraim, N. Merhav, Hidden Markov processes, IEEE Transactions on Information Theory 48 (6) (2002) 1518–1569.

[24] J.A. Hanley, The robustness of the "binormal" assumptions used in fitting ROC curves, Medical Decision Making 8 (3) (1988) 197–203.

[25] C. Metz, Basic principles of ROC analysis, Seminars in Nuclear Medicine 8 (1978) 283–298.

[26] F.J. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (3) (2001) 203–231.

[27] J. Daugman, Biometric decision landscapes, Technical Report UCAM-CL-TR-482, University of Cambridge, UK, 2000.

[28] M.A. Black, B.A. Craig, Estimating disease prevalence in the absence of a gold standard, Statistics in Medicine 21 (18) (2002) 2653–2669.

[29] K. Venkataramani, B. Kumar, Role of statistical dependence between classifier scores in determining the best decision fusion rule for improved biometric verification, Multimedia Content Representation, Classification and Security 4105 (2006) 489–496.

[30] J. Neyman, E.S. Pearson, On the problem of the most efficient tests of statistical hypotheses, Royal Society of London Philosophical Transactions Series A 231 (1933) 289–337.

[31] C. Shen, On the principles of believe the positive and believe the negative for diagnosis using two continuous tests, Journal of Data Science 6 (2008) 189–205.

[32] S. Thomopoulos, R. Viswanathan, D. Bougoulias, Optimal distributed decision fusion, IEEE Transactions on Aerospace and Electronic Systems 25 (5) (1989) 761–765.
[33] P.K. Varshney, Distributed Detection and Data Fusion, Springer-Verlag, New York, 1997.
[34] M.S. Pepe, M.L. Thompson, Combining diagnostic test results to increase accuracy, Biostatistics 1 (2) (2000) 123–140.
[35] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms Wiley, Hoboken, NJ, 2004.
[36] S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, Review of classifier combination methods, in: H.F. Simone Marinai (Ed.), Studies in Computational Intelligence: Machine Learning in Document Analysis and Recognition, Springer, 2008, pp. 361–386.
[37] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[38] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: ICML 96, 1996, pp. 148–156.
[39] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.
[40] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2) (2003) 181–207.
[41] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Journal of Information Fusion 6 (1) (2005) 5–20.
[42] R. Banfield, L. Hall, K. Bowyer, W. Kegelmeyer, A new ensemble diversity measure applied to thinning ensembles, in: Multiple Classifier Systems, vol. 2709, 2003, pp. 306–316.
[43] D. Ruta, B. Gabrys, Classifier selection for majority voting, Information Fusion 6 (1) (2005) 63–81.
[44] J. Kittler, Combining classifiers: a theoretical framework, Pattern Analysis & Applications 1 (1) (1998) 18–27.
[45] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.
[46] F. Roli, G. Fumera, J. Kittler, Fixed and trained combiners for fusion of imbalanced pattern classifiers, in: Proceedings of the Fifth International Conference on Information Fusion, vol. 1, 2002, pp. 278–284.
[47] T.K. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1) (1994) 66–75.
[48] M. Van Erp, L. Schomaker, Variants of the borda count method for combining ranked classifier hypotheses, in: Seventh International Workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000.
[49] D. Ruta, B. Gabrys, A theoretical analysis of the limits of majority voting errors for multiple classifier systems, Pattern Analysis & Applications 5 (4) (2002) 333–350.
[50] å. Raudys, F. Roli, The behavior knowledge space fusion method: analysis of generalization error and strategies for performance improvement, in: Multiple Classifier Systems, vol. 2709, 2003, pp. 55–64.
[51] D. MacKay, Ensemble learning for hidden Markov models, Technical Report, Cavendish Laboratory, Cambridge, UK, 1997.
[52] K. Tan, R. Maxion, Determining the operational limits of an anomaly-based intrusion detector, IEEE Journal on Selected Areas in Communications 21 (1) (2003) 96–110.
[53] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.
[54] J. Huang, C. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.
[55] D.D. Zhang, X.-H. Zhou, D.H. Freeman Jr., J.L. Freeman, A non-parametric method for the comparison of partial areas under ROC curves and its application to large health care data sets, Statistics in Medicine 21 (5) (2002) 701–715.
[56] E.M. Dos Santos, R. Sabourin, P. Maupin, Pareto analysis for the selection of classifier ensembles, in: Genetic and Evolutionary Computation Conference (GECCO), Atlanta, GA, USA, 2008, pp. 681–688.

About the Author—WAEL KHREICH is a PhD student in the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA) at the École de technologie supérieure (ETS). His main research interests are on-line and incremental learning for stochastic models such as hidden Markov models, and decision fusion in multi-classifier systems, with applications in intrusion detection in computer and network security.

About the Author—ERIC GRANGER obtained a PhD in Electrical Engineering from the École Polytechnique de Montréal in 2001, and from 1999 to 2001, he was a Defence Scientist at Defence R&D Canada in Ottawa. Until then, his work was focused primarily on neural network signal processing for fast classification of radar signals in Electronic Surveillance (ES) systems. From 2001 to 2003, he worked in R&D with Mitel Networks Inc. During that time, he designed algorithms and dedicated electronic circuits (ASIC/SoC) to implement cryptographic functions in Internet Protocol (IP)-based communication platforms. In 2004, Dr. Eric Granger joined the ÉTS, where he has been developing applied research activities in the areas of machine learning, patterns recognition, artificial neural networks, signal processing and microelectronics. He presently holds the rank of Assistant Professor in the département de génie de la production automatisée (GPA). Since joining ÉTS, he has been a member of the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), and his main research interests are adaptive classification systems, incremental learning, ambiguity and novelty detection, neural and statistical classifiers, and multi-classifier systems, with applications in military surveillance (recognition of radar signals), biometric authentification (recognition of individuals from their signatures and their faces), and intrusion detection in computer and network security.

About the Author—ALI MIRI is a Professor at the School of Computer Science, Ryerson University, 243 Church Street, Toronto, ON, Canada. He is also a Professor at the School of Information Technology and Engineering, and the Department of Mathematics and Statistics at the University of Ottawa, Ottawa, Canada. His research interest include applied cryptography, digital communication, signal processing, distributed systems, and mobile computing. He is a member of Professional Engineers Ontario, ACM and a senior member of IEEE.

About the Author—ROBERT SABOURIN joined in 1977 the Physics Department of the Montreal University where he was responsible for the design, experimentation and development of scientific instrumentation for the Mont Mégantic Astronomical Observatory. His main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montréal where he co-founded the Department of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he joined also the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil) where he was co-responsible for the implementation in 1995 of a master program and in 1998 a PhD program in applied computer science. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University).

Dr. Sabourin is the author (and co-author) of more than 260 scientific publications including journals and conference proceeding. He was co-chair of the program committee of CIFED'98 (Conférence Internationale Francophone sur l'Écrit et le Document, Québec, Canada) and IWFHR'04 (9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan). He was nominated as Conference co-chair of ICDAR'07 (9th International Conference on Document Analysis and Recognition) that has been held in Curitiba, Brazil, in 2007.

His research interests are in the areas of handwriting recognition, signature verification, intelligent watermarking systems and bio-cryptography.