Classification of Seismic Signals by Integrating Ensembles of Neural Networks

Yair Shimshoni and Nathan Intrator

***Abstract*—We examine a classification problem in which seismic waveforms of natural *earthquakes* are to be distinguished from waveforms of man-made *explosions*.**

We present an *integrated classification machine* (ICM), which is a hierarchy of artificial neural networks (ANN's) that are trained to classify the seismic waveforms. In order to maximize the gain of combining the multiple ANN's, we suggest construction of a redundant classification environment (RCE) that consists of several "experts" whose expertise depends on the different input representations to which they are exposed. In the proposed scheme, the experts are ensembles of ANN, trained on different Bootstrap replicas. We use various network architectures, different time–frequency decompositions of the seismic waveforms, and various smoothening levels in order to achieve an RCE. A confidence measure for the ensemble's classification is defined based on the agreement (variance) within the ensembles, and an algorithm for a nonlinear integration of the ensembles using this measure is presented.

An implementation on a data set of 380 seismic events is described, where the proposed ICM had classified correctly 92% of the testing signals. The comparison we made with classical methods indicates that combining a collection of ensembles of ANN's can be used to handle complex high dimensional classification problems.

***Index Terms*— Averaging, bootstrap, classification, combining estimators, ensembles.**

I. Introduction

WE EXAMINE a two-class classification problem in which seismic recordings of *natural* earthquakes are to be distinguished from the recordings of *artificial* explosions. This classification problem began to draw attention in the days of the cold war, when seismologic recordings were used to trace nuclear test explosions [3].[1] In recent years, researchers have addressed this problem using various disciplines other than classic seismologic methods, like artificial intelligence [4], pattern recognition [5], [6] and artificial neural networks (ANN's) [7], [8].

The vast majority of the recorded seismicity in most countries is artificial (i.e., man-made events like quarry blasts, mine explosions, military activity, road constructions, etc.). As the natural events are more important for the analysis of the regional seismicity and for seismic hazard assessment, constructing a reliable automatic method for selecting the signals of the natural events is crucial for efficient seismic research. Most of the work done thus far on this classification problem is concerned with regional events and nuclear explosions with magnitudes around 4 on the Richter scale, such as that in [3] and [7], rather than with local events and conventional explosions at near distances (magnitudes ≤ 2.5), although these are the majority of the artificial events being recorded. Many of the methods that appear in the literature [8], [9] are based on geophysical parametric models that need explicit information to be extracted or estimated by the analyst and, thus, are very difficult to be fully automated. Usually, the methods are based on local characteristics of the seismic activity rather than proposing general solutions.

The signal space formed by the seismic waveforms is very high dimensional and when dealing with weak local events, we have to face also a low signal-to-noise ratio due to the low signal energy. The nonhomogeneous crust of the earth and the various source mechanisms of seismic events cause the signal space to be complex and nonstationary.

In Section II, we specify a model based on ensembles of ANN's that can be applied for automated classification of all kinds of seismic events including weak local events in any geographical region. The model is data-driven and does not require prior geophysical information or any intervention of a human analyst. A multiexpert scheme is designed to handle the low energy and nonstationary nature of the signals. Given a recorded signal, we use different decompositions of its frequency spectrum in order to produce a class label i.e., *natural* or *artificial* event. Section III elaborates on the preprocess and implementation details followed by evaluation and results in Section IV.

Although it is implemented here on seismic classification, the proposed model is a general-purpose classification scheme that can be applied to a wide range of signal classification problems with two classes, whereas problems with more than two classes require only slight changes.

II. The Model

A. Combining Multiple Estimators

The lack of *a priori* knowledge about the true underlying model of the data in the seismic classification problem, like in many other real-life problems, leads the practitioners to examine various suboptimal classifiers. Different classifiers can exploit various types and sets of features; hence, combining multiple estimators might yield better performance than

Manuscript received February 15, 1997; revised November 30, 1997. The associate editor coordinating the review of this paper and approving it for publication was Prof. Yu-Hen Hu.

Y. Shimshoni is with the School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel.

N. Intrator is with the Department of Physics and the Institute for Brain and Neural Systems, Brown University, Providence, RI 02912 USA, on leave from the Computer Science Department, Tel-Aviv University, Tel-Aviv, Israel.

Publisher Item Identifier S 1053-587X(98)03270-X.

[1] Currently, seismologic recordings are used by the United Nations to monitor the compliance to the Comprehensive Test Ban Treaty (CTBT).

the best single candidate. In the neural network and machine learning literature, there are several methods for combining estimators, and the questions involved with this topic—such as *what* types of estimators to combine and *how* to combine them—have recently been getting considerable attention.

A well-known branch of the combined models contains methods like adaptive mixture of experts (AME) [10] and hierarchical mixture of experts (HME) [11], which are both based on the *divide-and-conquer* approach, where a mixture of experts competes to gain responsibility in modeling the output in a portion of the input space. The system's output is obtained as a linear combination of the experts' output, where the weights are computed as a parametric function of the inputs by a gating module. The underlying probabilistic model is based on the assumption of mutual exclusivity, i.e., a single expert is responsible for each data point. In the mixture models like AME and HME, the different experts are usually trained on a single data set simultaneously by minimizing a combined cost function. The final combination of the experts is determined by the gating module that is constructed during the same training session. When training all the experts on the same data set with the same data representation, the dependence of errors among experts is high, thus diminishing their collective contribution.

A different aspect of the multiple model concept is introduced by using a committee of classifiers, which are also called *ensemble* [12], [13]. Consider a trained classifier as a realization of the generic model trained on the given data. Thus, different data sets would yield different realizations, all of which are members of the "*post training*" distribution of the possible solutions. As the solution space is generally highly degenerate and includes many local minima, it is more robust to use a sample from this solution space (i.e., ensemble of estimators) rather than a single representative [14]. The ensemble of classifiers can be averaged to produce an aggregated classifier, or any linear combination of the realizations can be used.

When all the classifiers give similar results, the accuracy of their combined classification depends largely on their bias due to the bias-variance tradeoff [15]. Whenever the bias of a generic model is high, the multiple classifications of its ensemble might all be wrong even if the variance is low; then, no combination will help. On the other hand, when the bias is small and the variance is high, we can expect the ensemble to disagree whenever the input signals are ambiguous. In such cases, the ensemble's result, i.e., the aggregated classification, will have the same bias but a reduced variance. Hence, combining multiple classifiers can eliminate the need to regularize overfitted models with high variance [16]. Moreover, the classification confidence can be evaluated from the variance that represents the agreement within the ensemble, where signals with high classification variance are treated with suspicion.

Estimating the optimal combination of the experts should be done in a robust way, i.e., by averaging or cross validation techniques rather than by parametric estimation based on the same training data [17]. It has been shown that in order for the combination of experts to be optimal, the experts should be made as independent as possible [2], [18], [19].

Another method that uses multiple classifiers in a sequential training scheme is boosting [20], [21]. In this method, which is typically suitable for very large training sets like in optical character recognition (OCR) problems, each classifier is trained on patterns that have been filtered by the previous classifier. Thus, the result is a combination of classifiers that were trained on statistically different data sets in a sequential process.

A more general framework for combining multiple estimators is "stacked generalization" [22], where each estimator is trained with a different subset of the data, and the optimal combination is estimated using cross-validation methods. A formulation of this method for regression estimators was presented in "stacked regression" [23] and compared with other methods in [17].

Bagging is a method that produces an aggregated estimator using bootstrap replicas of the training data [1]. It is reported to be useful whenever the estimator is unstable, i.e., when perturbing the training set can cause significant changes in the constructed classifier. Notice that this condition corresponds to the requirement mentioned earlier of maximum independence among the experts. Several ways were suggested for making the experts less dependent; one example is to inject noise during the training, as in smooth bootstrap [24].

Since the search for an optimal classifier is tied with the search for an optimal data representation, i.e., an optimal transformation of the input signals w.r.t the classification task at hand, it is advisable to examine and possibly use more than one signal representation. In order to maximize the information extracted from the example data and the gain of combining multiple classifiers, we suggest to construct a redundant classification environment (RCE), which allows for different signal representations to supply a wide coverage of the effective feature space.

B. Creating a Redundant Classification Environment

The hierarchical scheme that is proposed, is based on experts that are ensembles of ANN's, each of which is associated with a specific setting of data representation and network architecture. The redundancy is pronounced within these ensembles, which are collections of ANN realizations trained on different subsets of the data. Each ensemble is trained using the same data representation, i.e., a unique time–frequency decomposition and smoothening level of the input signals. The network architecture (number of hidden units) is also fixed for all members in an ensemble. The redundancy is pronounced even further, as we use various combinations of time–frequency decompositions, smoothening levels, and network architectures to create and train several such ensembles. The different training conditions yield relatively independent classifiers that might produce different classification results given an ambiguous signal. The *integrated classification machine* (ICM) that is formulated next integrates the different ensembles in this classification environment and produces a final classification that achieves better generalization performance than the single classifying components (see the results in Section IV).

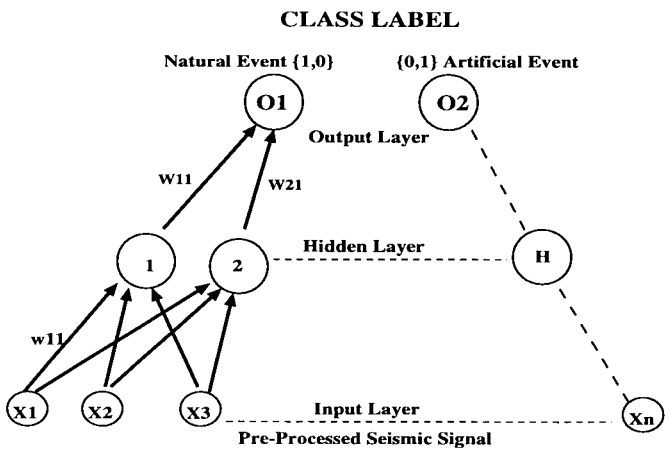


Fig. 1. Feed-forward neural network with input dimensionality N and one hidden layer with H units and two output units as used for classification in the ICM.

C. The Integrated Classification Machine

The ICM is constructed of a hierarchy of classifiers, as shown in Fig. 1. The smallest building block of the ICM is a neural network known as the feedforward multilayer perceptron with sigmoidal activation functions: $\sigma(c) = (1/(1+e^{-c}))$; see Fig. 1). All networks are trained to predict the class label of a given seismic signal (Section III-C gives the training details). As mentioned, we use several representations for the input signals; thus, the input layer of the neural networks in the ICM has different dimensionalities (N) according to the respective input representation used. The hidden layer can contain various numbers (H) of sigmoidal units, and the output layer contains two sigmoidal units.

1) The Network's Prediction Value: The desired output of the networks for a given signal can be either $\{1, 0\}$ for *natural* events or $\{0, 1\}$ for *artificial* events. The sigmoidal output units of the trained networks will produce continuous values in the range of 0–1 according to the network weights

$$O^l = \sigma\left(\sum_{i=1}^{H} W_{li}\sigma\left(\sum_{j=1}^{N} w_{ij}x_j + w_{i0}\right) + W_{l0}\right) \quad l = 1, 2. \tag{1}$$

Let us define the (signed) difference of the two output units $y = (O^1 - O^2)$ as the *prediction-value* of the network. Hence, $y \in [-1, 1]$, and the predicted class label is given by thresholding y at zero assigning the positive values to the class of *natural* events and negatives to the class of *artificial* events. Each network is trained on T repeated trials,[2] changing only the initial random weights. We define the *prediction-value* of the network component in the ICM w.r.t. a signal x as the average *prediction-value* $y(x)$ of these T training trials $y^{\mathrm{NET}}(x) = (1/T)\ \Sigma_{t=1}^{T}\ y_t(x)$.

2) The Ensemble's Prediction Value: Each ensemble is a collection of B networks, where each network is trained on one of the B replicas[3] of the original data set $D_r^b b = 1, \cdots, B$ (for a specific data representation r). Details on the replication of the data into bootstrap sample sets [25] are given in Section III-B.

All the networks in an ensemble share the same data representation and the same network architecture. The *prediction-value* of an ensemble w.r.t. a signal x is defined as the average over all the *prediction-values* of the participating networks, as in the "bagging" method [1]

$$y^{\mathrm{ENS}}(x|D_r) = \frac{1}{B}\sum_{b=1}^{B} y_b^{\mathrm{NET}}(x|D_r^b). \tag{2}$$

3) The Integrated Prediction Value: A collection $(\mathcal{K})$ of ensembles, which use different input representations, i.e., time–frequency decompositions or smoothening levels, and different network architectures (number of hidden units) form the ICM shown in Fig. 2. The integrated *prediction-value* of the ICM w.r.t. a signal x is defined as

$$y^*(x) = \sum_{k\in\mathcal{K}} \alpha_k\beta_k(x)y_k^{\mathrm{ENS}}(x) \tag{3}$$

where α_k is a prior reliability measure of the kth ensemble, which can be determined from the training data or assumed using prior knowledge (otherwise, $\alpha_k = (1/K)$). The second value $\beta_k(x)$ is a posterior classification confidence measure that is specific to each signal and is discussed in the next section. Both measures are normalized and determine together the strength of the vote of ensemble k in the classification committee of signal x.

A. Measuring the Classification Confidence

The seismic signals of the artificial and natural classes are separated along the high-dimensional input space in a very nonsmooth manner; thus, it is reasonable to assume that incorrect classifications will occur more often for signals located in the vicinity of decision boundaries. The difficulty in classifying a given signal can be measured by its distance from these boundaries. Unfortunately, in a high-dimensional nonlinear input space, as in the current situation, it is not realistic to estimate these boundaries explicitly. In order to detect signals with ambiguous class membership and to rank the different ensembles by their accuracy of classification, we have constructed a confidence measure for the ensemble's classification. This posterior confidence is based on the variance of the networks' *prediction-value*

$$\mathrm{CONF}^{\mathrm{ENS}}(x) = [\mathrm{VAR}(y^{\mathrm{NET}}(x))]^{-1} \tag{4}$$

where $y^{\mathrm{NET}}(x)$ is the network's *prediction-value*. The CONF score represents the amount of "agreement" among all the participating networks in the ensemble [2], [14]. When the bias of the networks is high, such a measure will not convey the desired confidence score, i.e., when all members agree on the same wrong class. In our work, we are dealing with neural networks with high input dimension and reasonable capacity. Moreover, neural networks are considered unstable estimators [1] and are known to suffer more from variance than from bias. Still, one should examine the classification results carefully to

[2] In our implementation, $T = 5$.

[3] We used 30 bootstrap sets, which is less than suggested in [25] (≈ 200) but was found to be sufficient.

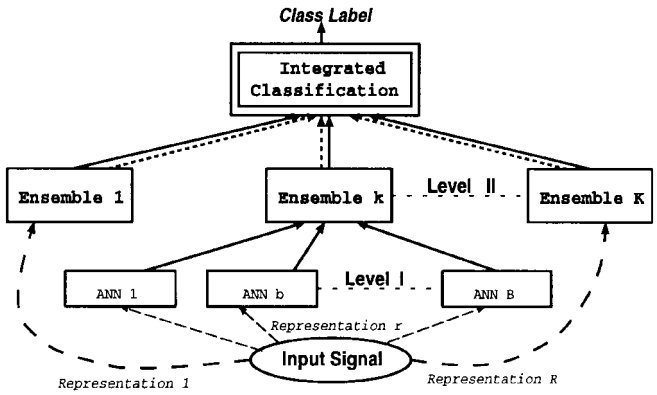


Fig. 2. Integrated classification machine. Several representations of the waveform are fed into different ensembles, then integrated to produce the final classification. (In level II, the regular arrows are the ensembles' prediction values, and the dashed arrows are the attached confidence values).

check whether there is a correlation between the confidence scores and the errors of the combined classification.

E. Combining the Hierarchy of Classifiers

The ICM shown in Fig. 2 is a hierarchy of classifiers with two levels of combining multiple classifiers. In the first level, for each ensemble, B networks are combined using simple averaging to construct an aggregated ensemble classification (2). These networks are in fact multiple realizations of the same classifier trained on different bootstrap samples. This method for combining estimators is called "bagging" (see Section II-A).

The second level of combining multiple classifiers in the ICM is the integration of ensembles into the final classification (3). When integrating the ensembles, a decision must be made with regard to the strength of the "vote" of each ensemble in the classification of a given signal x, i.e., the $\beta_k(x)$'s in (3) must be estimated. In order to do that, we can apply fixed weighting methods like simple averaging of the ensembles, as we have done in the first level, or apply least-squares combination with nonnegativity constraint [23]. Finally, prior weights can be used as well to combine the ensembles in a fixed manner. Uunlike the network integration into ensembles at the first level, where each network is the same apart from being trained on a different subset of the data, in level II, it is less likely that a fixed weighting will produce better classification results than *all* single ensembles. There might be some ensembles that use inferior data representations or less suitable model architectures, which will have disturbing effects on the weighted result. In practice, we have noticed that for fixed weighting methods like the ones mentioned before, the integrated classification results often were worse than the best participating classifier (ensemble).

Given a low signal-to-noise ratio and nonstationarity of the signal space, along with shortage of training data, it is furthermore desirable not to decide on the integration coefficients, based on the characteristics of the currently available data set. Therefore, we suggest using the signal x to find the optimal integration of its own "classification committee," namely, to apply a nonfixed weighting of the classifiers. One can use the ensemble's *prediction value* $y^{\mathrm{ENS}}(x)$ as the basis for the weighting or decide the strength of the votes using the classification confidence $\mathrm{CONF}^{\mathrm{ENS}}(x)$ [26].

We have used a nonfixed weighting strategy that is a dynamic "*winner takes all*" selection. In order to integrate the different ensembles in the ICM for a *specific* signal x, all classifiers (ensembles) are ranked, and the optimal *one* is selected. The ranking is by the overall reliability of the classifiers, and the selection is based on the classification confidence $\mathrm{CONF}^{\mathrm{ENS}}(x)$ supplied by the ensembles along with their classification (see Fig. 2). This approach, which will be elaborated upon in the next section, is different from the linear nonfixed weighting that was suggested by [26] for combining ANN's, where all classifiers participate in the committee with weights inversely proportional to the variance (which is similar to the CONF values used here but estimated in a different way as they use single ANN classifiers rather than ensembles).

By applying this method, we aim to exploit both the robustness of the aggregated classification of the ensembles and the adaptiveness of our integration strategy, which selects the most suitable ensemble for each signal out of the entire classification environment.

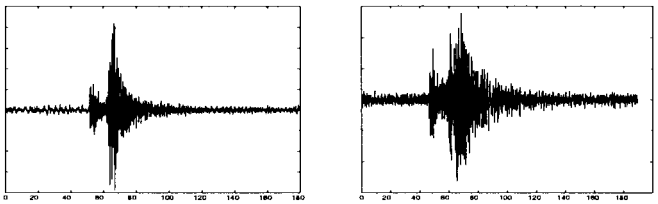
F. Competing Rejection Algorithm (CRA)

Generally, a signal is said to be rejected by a classifier if some measure representing the quality of its classification does not fall below a predefined threshold. Obviously, the higher the threshold, the more signals will be rejected, and thus, the smaller the misclassification rate will be over the remaining unrejected signals. We present an algorithm that performs a sequence of classifications by polling the group of K classifiers w.r.t. the signal at hand. Each classifier (ensemble), in turn, can either classify or reject the signal. The main motivation is based on the observation that some classifiers perform globally better than others. Nevertheless, classifiers can outperform "superior" classifiers on a local basis and, thus, should be given the opportunity to compete and possibly "steal" a classification whenever the signal was rejected by those "superior" classifiers.

In order to implement this idea, a prior reliability ranking of the classifiers has to be set, and a rejection criterion has to be defined. The rejection is done by thresholding the confidence measure $\mathrm{CONF}^{\mathrm{ENS}}(x)$ (4), i.e., a classification is rejected when its confidence value is lower than the *reject-threshold*. Each ensemble k has its own threshold, depending on its accuracy and variability that fixes the minimum level of confidence "allowed" for its classifications. The *reject-threshold* is calculated as a certain upper percentile of the confidence scores $\mathrm{CONF}^{\mathrm{ENS}}(x)$ of an unlabeled data set D^* as

$$\begin{aligned} &\mathit{Reject\text{-}Threshold}_k \\ &\quad = \text{percentile}\ \{\mathrm{CONF}^k(x),\ \mathit{Reject\text{-}Rate}_k\}. \end{aligned} \tag{5}$$

The *reject-rate* of a classifier represents its global credibility and can be determined from the performance on training data or by using subjective information. In the simple case, all ensembles are considered to have the same credibility, and



(a) (b)

Fig. 3. (a) Natural earthquake event. (b) Artificial explosion event. The horizontal axis is recording time in seconds (The sampling rate is $50\ H_z$).

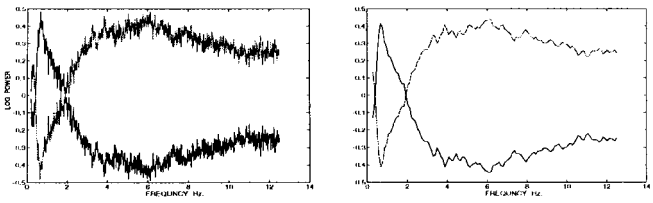


Fig. 4. Class means on the left are for standardized log power-spectrum (dim $= 1024$) of *W1*. On the left the class means for the smoothened representation (dim $= 127$). Notice that the differences between classes are smaller than the STD (unit) of the data set. (The reflection of the mean spectra is caused by the similar size of the classes in the data set.)

a uniform *reject-rate* is set (for example, the upper 20% percentile). When all classifiers reject a signal, it can be either "globally rejected" or classified by the *ultimate classifier*, which is optionally predefined by the user. The global rejection rate cannot be determined by the user and is usually much smaller than the single *reject-rates*.

Notice that this selection algorithm and the whole ICM structure is easily scalable, and no retraining is required when new classifiers are added. It is also flexible in the sense that it is straightforward to incorporate other types of experts (including human ones), as long as they produce suitable *prediction-value* and CONF value.

III. Implementation for Seismic Classification

A. Preprocessing the Seismic Signals

The proposed ICM was implemented on a comprehensive data set[4] consists of 380 seismic events, which includes all the natural local earthquakes that occurred from January 1990 to June 1993 inside an area of 22 500 km^2 in the north part of Israel. A similar number of artificial explosion events were randomly sampled from the same spatio-temporal window. All events were recorded by a vertical component short-period seismometer (station JVI of the Israeli Seismic Network). The recorded signals are bandpass filtered to a frequency band (0.2–12.5 Hz) and transmitted via FM telemetry to the station, where they are digitized with sampling rate of 50 Hz using a 12-bit A/D converter (see Fig. 3). All events have magnitude $M_L < 2.7$, whereas 77% of the events are below 2.0, and the mean magnitude is 1.53.

The recorded seismic signals are mixed with several types of noise, such as station background noise and telecommunication interferences. The underground path through which the seismic waves travel (on their way from the source to the seismometer) also has a considerable effect on the waveform. A given recorded signal includes 8000 samples on average, about half of which consist of the actual seismic event, whereas the rest are recorded before and after the event for synchronization with other seismometers. We cut those samples by detecting the signal onset (with an automatic procedure[5]) and taking a fixed size window of ~ 2000 samples starting at the onset. The window is about 45 s of recording [27]. We have used three spectral decompositions of the waveforms for our basic input representations (denote *W1, W3, W10)*. The first frequency decomposition is made on a single window of 2048 samples starting from the signal's onset. The second is based on three windows, each with 1024 samples and 384 overlapping samples. For the last representation, we have used ten windows of 256 samples and 32 overlapping. The size of those windows and their overlapping defines the *time–frequency* resolution of the representation. Notice that we preferred to keep all three representations about the same dimensionality. Obviously, if the segmentation in time is finer, then the spectral decomposition will get coarser. However, we can also enlarge the overlapping factor; thus, larger windows can be used, and finer spectral resolution is gained at the expense of less locality in the time representation. In such a case, some resolution w.r.t. the changes through time is still partially maintained, but then, we pay the crucial cost of a great expansion in the dimensionality of the representation, which is undesirable in our case, as it amplifies the "*curse of dimensionality*" [28].

After the FFT was applied on a window of samples, it consists of frequency coefficients that represent the amplitude of the respective frequency components in this specific window. As the seismic power spectra show a fast decay of energy, it was more appropriate to use a *logarithmic* scale, which also has a smoothening effect on the signals. In the *W3* and *W10* representations, the transformed windows were concatenated to form the final input representation. The signals after the log scaling were standardized making each of the dimensions (frequency coefficients) zero-mean with variance equal one unit. Fig. 4(a) shows the standardized means for the two classes of the *W1* representation (dim $= 1024$). The preprocessed signals contain more than 1000 dimensions and are hardly smooth. Before training any classifier on the data set, we smoothen it (thus reducing its dimension) by taking the averages of a sliding window.

We have applied four levels of smoothening, achieving final dimensionalities of 333, 127, 75, and 31. (Fig. 4 shows the class means after smoothening). The nature of the preprocess procedure done here (FFT, scaling, standardization, and smoothening) is straightforward and is common in many other domains of problems like speech recognition, etc. We might use sophisticated preprocessing of the seismic signals to equalize the effect of several factors on the recorded signals, i.e., the distance from the source to the seismometer or the event's magnitude [7]. We have decided to stay within the common preprocess procedures in order to avoid an additional bias in our model and to keep it as general as possible.

[4]The seismic data set is available via ftp://www.math.tau.ac.il/~shimsh/pub/seismic-data/

[5]Details on our detection algorithm are on p. 5 of ftp://www.math.tau.ac.il/~shimsh/pub/refs/results.ps.Z

B. Constructing the Bootstrap Sample Sets

Eight input representations were selected for the original data set D (combining three time–frequency resolutions and four smoothening levels). For each one, we have constructed a data set $D_r, r = 1, \cdots, 8$, containing $N_1 = 191$ *earthquakes* and $N_2 = 189$ *explosions*. Taking the distribution $\mathcal{F}_D$ of the given data as a plug-in estimator of the real distribution of the data $\mathcal{F}$, we followed the "*bootstrap pairs*" technique [17].

The original data set was pseudo replicated B times in the following way: Let $\{I_i^c\}_{i=1}^{N_c}$ be sets of indices for the two classes $c = 1, 2$. Construct B *bootstrap sample sets* of size $(N_1 + N_2)$ by sampling uniformly with repetitions from the two sets of indices, respectively [notice that both classes should be sampled separately to preserve the class probability $P(c_i)$ in the resulting replicated samples]. The resulting B different TRAIN-SET's consist of 380 indices and are partially overlapping as they contain multiple appearances of some indices (hence, they do *not* contain about 36% of the indices on average). The probability that an index will be excluded from one replicated set of size N is $(N - 1/N)^N$ asymptotically equals to 0.368. In our case, where $N_1 = 191; N_2 = 189$ and $B = 30$, the expected number of TRAIN-SET's (bootstrap sample sets) from which a signal will be excluded is about 11. In other words, each replicated TRAIN-SET contains approximately 240 distinct indices. The remaining (unsampled) indices form a corresponding TEST-SET for each of the B TRAIN-SET's. These sets are *not* used in any part of the training process or model selection; they are used only to evaluate the generalization performance of the model.

In conclusion, for each input representation, we have the data sets D_r replicated into 30 couples of disjoint sets such that $D_r^b = \text{TRAIN}_r^b \cup \text{TEST}_r^b, b = 1, \cdots, 30$. The sampled indices in each of those bootstrap sample sets are the same for all D_r's, regardless of the representation; therefore, we can accurately compare classifiers of different representations.

C. The Networks' Training Scheme

All the networks were trained using gradient-based least-squares learning with the back-propagation algorithm[6] [29]. We used a fixed low learning rate $\eta = 0.001$. The initializing weights were sampled from a uniform distribution $\text{U}[-0.1, 0.1]$. We used a *batch-type* training, and the number of epochs was limited to 1000 in order to avoid exhaustive learning of the training data.

IV. Results

A. Evaluation of Performance

The performance estimates we used for comparison and evaluation of the classifiers were based on cross-validation techniques [25]. The networks at the bottom of the ICM structure are evaluated by averaging over $T = 5$ trials. A single network's performance is estimated via the misclassification rate over the TEST-SET, which corresponds to the TRAIN-SET on which the network was trained. The performance of an ensemble can be evaluated by averaging the performance of its 30 networks, and the final classification can be evaluated accordingly by averaging over the ensembles. This simple bottom-up evaluation does not take into consideration the *combined* classification, which may improve the average result [1]. This error evaluation is therefore used only as an indicator for the ICM's error rate.

In the setting we defined, the misclassification rate (MCR) of the combined ensemble *cannot* be measured over the whole data set because it will not be a pure cross-validated result. In order to present a cross-validated error estimation that is more realistic than the simple average of the MCR's, we classify each signal x by combining only a subset of the ensemble, namely, only those networks that were trained on bootstrap samples sets that did not contain the signal x. Denote this subset of networks as the *cross-validated subset* $CV(x)$ corresponding to signal x. It consists of $B^* < B$ networks (as described before, each signal was excluded from about 11 TRAIN-SET's). Now, we can produce a combined classification for each signal in the original data set, based on the B^* networks in the corresponding $CV(x)$. The cross-validated *prediction-value* is defined as

$$\hat{y}^{\text{ENS}}(x) = \frac{1}{B^*} \sum_{b \in CV(x)} y_b^{\text{NET}}(x). \tag{6}$$

This estimate [25] is denoted as ϵ_0 and is used in a weighted combination with the apparent error (which is calculated over the original data set) to construct the 0.632 estimator[7]: $\text{Err}^{.632} = 0.632\epsilon_0 + 0.368\ \text{Err}_{\text{app}}$. We think that in the framework of trained neural networks, the 0.632 estimator is still quite optimistic; therefore, we have used the pure ϵ_0. Notice that in a later work, the 0.632 estimator was refined to enable more freedom in choosing the weights of the Err_{app} and the ϵ_0, as a function of the *relative overfitting rate* [30].

As we are combining only a third of the available classifiers in each ensemble and use the (possibly pessimistic) estimate ϵ_0, we suggest that we should consider our reported evaluation of the ensembles' performance (and, thus, of the whole ICM) as an upper bound of its true performance.

B. Comparing the Different Ensembles

The ICM model we presented in this work was tested on a comprehensive seismic data set consists of 380 events. Fifteen ensembles (30 networks each) were trained with five trials on eight different input representations with various numbers of hidden units (2250 ANN's in total). The data set along with the complete results are available via `ftp` [31].

The results suggest that the most relevant factor is the representation, mainly, the time–frequency resolution as well as the smoothening level (input dimension). Out of the three time–frequency resolutions, the one with the single window *W1* achieved the best results. This finding might suggest that information on the spectral changes through time has less

[6] We have used a customized version of TRAINBP function from the neural network toolbox of MATLAB 4.2.

[7] The 0.632 corresponds to the probability for a signal to be included in a bootstrap sample set (see Section III-B).

TABLE I
MISCLASSIFICATION RESULTS FOR DIFFERENT METHODS. LEFT: AVERAGES OF SINGLE CLASSIFIERS. RIGHT: COMBINED CLASSIFICATION RESULTS. (k-NN: k-NEAREST NEIGHBOR, LDA: LINEAR DISCRIMINANT ANALYSIS, ANN: ARTIFICIAL FEED-FORWARD NEURAL NETWORK)

Rep.		Single Classifier			Aggregated Classifier		
T-F	DIM	*k-NN*	*LDA*	*ANN*	*k-NN*	*LDA*	*ANN*
W1	333	0.141	0.462	0.099	0.116	0.437	0.082
W1	127	0.137	0.215	0.100	0.116	0.174	0.087
W1	50	0.129	0.132	0.102	0.113	0.121	0.089
W1	31	0.142	0.117	0.110	0.118	0.105	0.095
W3	375	0.156	0.467	0.104	0.137	0.392	0.097
W3	75	0.155	0.149	0.111	0.134	0.139	0.103
W10	300	0.191	0.484	0.125	0.176	0.471	0.111
W10	120	0.213	0.238	0.134	0.168	0.195	0.113

discriminant power. For all time–frequency resolutions, the larger models (higher input dimensionalities) yielded better results than models with smoothened data. This probably has occurred because of the lower bias of the large models due to their greater capacity [15]. The bias–variance calculation that appears in the complete results [31] showed that the massive averaging decreases the variance of the classifiers, thus eliminating the need for regularization.

Both for the single networks and for the aggregated ensemble results, the preferred model are of a single spectral window *W1* using the higher input dimensionality (DIM $= 333$). The uncombined results (level I) are improved after aggregation (by the ensemble) by 10–20% on average, proving the benefit of averaging multiple ANN's.

C. Comparison with Classical Methods

We have compared our ANN ensemble with two classical methods using the same training and testing sets (the bootstrap sample sets) and under the same scheme of aggregating multiple classifiers. The classifiers tested were *linear discriminant* and K*-nearest neighbors* (with several K values).

The results for the average misclassification rate (MCR) of single classifiers (30 realizations) and for the aggregated classification are shown in Table I. The results for *ANN* and k-NN are for the best among several values of H and k, respectively. The linear classifier was incapable of coping with the larger models and did not show the same scalability as the neural network models. In general, one can see that the *neural network* classifiers outperformed both the linear classifier and the K-NN classifier.

D. Results for the Integrated Classification

To test the performance of level II of the ICM, we have grouped the large models of the three time–frequency representations, providing a wide time–frequency coverage. The ensembles were ranked for the competing rejection algorithm at the order *W1-333, W3-375, W10-300,* and all three were with six hidden units. the *ultimate classifier* was set to be *W1-333*, and thus, all signals were eventually classified to one of the two classes. The *rejection rate* was set to 20% for all three ensembles, and the *rejection thresholds* were extracted from the unlabeled data, as described in Section II-F.

TABLE II
INTEGRATED AND NONINTEGRATED MISS-CLASSIFICATION RATES. THE FIRST COLUMN CORRESPONDS TO NONLINEAR INTEGRATION WITH THE COMPETING REJECTION ALGORITHM, THE SECOND COLUMN SHOWS THE RESULT WITH LINEAR INTEGRATION BASED ON VARIANCE, AND THE THIRD COLUMN REFERS TO FIXED UNIFORM AVERAGING. THE LAST THREE COLUMNS SHOW THE MCR FOR SINGLE (UNINTEGRATED) ENSEMBLES

Integrated Machine			Ensembles		
CRA	VAR	AVG	W1-333	W3-375	W10-300
0.079	0.095	0.095	0.082	0.097	0.113

Table II shows MCR's for three integration methods. The first is the competing rejection algorithm (CRA), which we have presented, and the other methods of integration are variance-based weighting (VAR) as suggested by [26] and uniform averaging (AVG).

The results show that the best ensemble's performance (based on *W1*, dim $= 333$) was further improved by the integration with the two inferior ensembles of the three and ten spectral windows *(W3-375, W10-300)* when the CRA was used. Integration with the other two methods has failed to reach the performance of the best single ensemble.

V. CONCLUSIONS

We have addressed the problem of seismic signal classification by constructing an *integrated classification machine* (ICM), which consists of a collection of neural networks' ensembles. A redundant classification environment (RCE) was created, by which we tried to achieve a robust classification for the noisy and nonstationary seismic signals.

Examining the classification results corresponding the different input representations, it appeared that there was no gain in using more than a single spectral window for the nonintegrated classification. Nevertheless, when ensembles from different time-frequency resolution were integrated by the competing rejection algorithm (CRA), there was a complementary affect yielding an improved performance.

The main factor influencing the classification performance was the input representation used, namely, the time–frequency resolution and the smoothening level. The best result for a single classifier was achieved using a single spectral window *W1* and low level of smoothening (DIM $= 333$).

For all time–frequency resolutions, the larger models (higher input dimensionalities) yielded better results than models with smoothened data.

It seemed that the main contribution to the classification error was due to the *bias* of the classifiers, which is not eliminated by the averaging.[8] Since the massive averaging reduces the classification variance, it is advisable in a multiple classifier model (like the ICM we have presented) to use

[8]The exact bias–variance calculation can be found on p. 27 of the results appendix, which is available from: ftp://www.math.tau.ac.il/~shimsh/pub/refs/results.ps.Z

classifiers with larger capacity, thus giving us greater ability to handle high-dimensional signals with lower bias.

Comparing the ANN with classical classification methods shows that ANN classifiers are preferable for the problem at hand over the LDA classifiers, which did not show the same scalability as the ANN models. The K-NN classifier showed some scalability but performed worse than the ANN for all representations.

Improved performance was achieved by integrating ensembles of ANN trained on different data sets or input representations using nonconstant weighting based on a classification confidence. We have used the classification variance as a posterior confidence measure by which the CRA selects the optimal classifier for each signal. By applying such a method, we aimed to exploit both the robustness of the aggregated classification (level I) and the adaptiveness of the integration strategy (level II). The integrated classification by this method outperformed the nonintegrated models as well as other integration methods like linear nonfixed weighting (also based on the variance) and uniform averaging of the ensembles. The best overall result achieved was 92% of correct classifications on testing data.

The ICM model is a general framework and can be applied for other waveform classification problems. The high performance achieved on the seismic classification at hand (being based on data from one seismometer only), motivates an implementation of the proposed method for classification of seismic events, which is a task still carried out by humans.

REFERENCES

[1] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, 1996.

[2] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," *Advances Neural Inform. Process. Syst.*, vol. 7, 1995.

[3] R. S. Taylor, M. Denny, E. Vergino, and R. Glaser, "Regional discrimination between NTS explosions and western US earthquakes," *Bull. Seismic Soc. Amer.*, vol. 79, pp. 1142–76, 1989.

[4] C. Chiaruttini, V. Roberto, and F. Saitta, "Artificail intelligence techniques in seismic signal interpretation," *Geophys. J. Int.*, vol. 98, pp. 265–73, 1989.

[5] M. Joswig, "Pattern recognition for earthquake detection," *Bull. Seismic Soc. Amer.*, vol. 80, pp. 170–186, 1990.

[6] J. Wüster, "Discriminate of chemical explosions and earthquakes in central Europe—A case study," *Bull. Seismic Soc. Amer.*, vol. 83, pp. 1184–1212, 1993.

[7] F. U. Dowla, S. Taylor, and R. Anderson, "Seismic discrimination with artificial neural networks: Preliminary results with regional spectral data," *Bull. Seismic Soc. Amer.*, vol. 80, pp. 1346–1373, 1990.

[8] P. S. Dysart and J. Pulli, "Regional seismic event classification at the NORESS array: Seismological measurements and the use of trained neural networks," *Bull. Seismic Soc. Amer.*, vol. 80, pp. 1910–1933, 1990.

[9] M. A. H. Hedlin, J. B. Minster, and J. A. Orcutt, "An automatic means to discriminate between earthquakes and quarry blasts," *Bull. Seismic Soc. Amer.*, vol. 80, pp. 2143–2160, 1990.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[11] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, pp. 181–214, 1994.

[12] L. K. Hansen and P. Salamon, "Neural networks ensembles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 993–1001, 1990.

[13] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble method for neural networks," in *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. New York: Chapman-Hall, 1993.

[14] L. K. Hansen, C. Liisberg, and P. Salamon, "The error-reject tradeoff," available via FTP from NeuroProse (`archive.cis.ohio-state.edu`), 1994.

[15] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias-variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, 1992.

[16] P. Sollich and A. Krogh, "Learning with ensembles: How over-fitting can be useful," *Advances Neural Inform. Process. Syst.*, vol. 8, 1996, to be published.

[17] M. LeBlanc and R. Tibshirani, "Combining estimates in regression and classification," available FTP from NeuroProse (`archive.cis.ohio-state.edu`), 1993.

[18] R. Meir, "Bias, variance and the combination of least squares estimators," *Advances Neural Inform. Process. Syst.*, vol. 7, 1995.

[19] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Comput.*, vol. 7, pp. 867–888, 1995.

[20] R. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, pp. 197–227, 1990.

[21] H. Drucker, C. Cortes, L. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensemble methods," *Neural Comput.*, vol. 6, pp. 1289–1301, 1994.

[22] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[23] L. Breiman, "Stacked regression," Tech. Rep. TR-367, Dept. Statistics, Univ. California, Berkeley, Aug. 1992.

[24] Y. Raviv and N. Intrator, "Bootstrapping with noise: An effective regularization technique," *Connection Sci.*, vol. 8, pp. 355–372, 1996.

[25] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman-Hall, 1993.

[26] V. Tresp and M. Taniguchi, "Combining estimators using nonconstant weighting functions," *Advances Neural Inform. Process. Syst.*, vol. 7, 1995.

[27] Y. Shimshoni and N. Intrator, "Classification of seismic signals by integrating ensembles of neural networks," in *Proc. Int. Conf. Neural Inform. Process.*, Hong Kong, 1996.

[28] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ: Princeton Univ. Press, 1961.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *PDP, vol. 1*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA; MIT Press, 1986.

[30] B. Efron and R. Tibshirani, "Cross-validation and the bootstrap: Estimating the error rate of a prediction rule," Tech. Rep. TR-477, Dept. Statistics, Stanford Univ., Stanford, CA, 1995.

[31] Y. Shimshoni, "Classification of seismic signals using ensembles of neural networks—Results appendix," 1995, available via (`ftp.math.tau.ac.il/pub/shimsh/results.ps.Z`).

Yair Shimshoni is a Ph.D. student at the computer science department at Tel-Aviv University, Tel-Aviv, Israel. His areas of interest include neural computation and machine learning, classification and prediction using methods for combining experts, and automation of seismic data analysis.

Nathan Intrator received the Ph.D. degree from Brown University, Providence, RI.

He currently holds positions at the Computer Science Department, Tel-Aviv University, Tel-Aviv, Israel, and at the Institute for Brain and Neural Systems, Brown University. He has contributed to the mathematical analysis of the Bienenstock, Cooper, and Munro (BCM) theory and applied his methods to various real-world applications including speech and 3-D object recognition. His areas of research include neural computation, high-dimensional statistics, methods for controlling bias and variance of predictors, combining experts and model discovery via neural networks.