A hybrid recorded-synthetic sonar data set for validation of ASW classification algorithms

Karl Thomas Hjelmervik
Norwegian Defence Research Establishment (FFI)
Horten, Norway
Email: karl.thomas.hjelmervik@ffi.no

Henrik Berg
Norwegian Defence Research Establishment (FFI)
Horten, Norway
Email: henrik.berg@ffi.no

Dan Henrik Sekse Stender
Norwegian Defence Research Establishment (FFI)
Horten, Norway
Email: dan-henrik.stender@ffi.no

Tale Solberg Såstad
Norwegian Defence Research Establishment (FFI)
Horten, Norway
Email: tale.solberg@ffi.no

***Abstract*—Modern anti-submarine warfare sonars are often designed with narrow beamwidths and wide frequency bandwidths in order to maximize spatial resolution and sonar performance. A known issue for high-resolution sonars in littoral environments, is the occurence of high false alarm rates. Increased false alarm rates increase the workload of sonar operators and also reduces the usefullness of automatic systems such as autonomous underwater vehicles, since their limited communication abilities hinder them from sharing large amounts of contacts.**

The false alarm rate may be reduced simply by increasing the threshold used in the detection process. However, this also reduces the probability of detecting actual targets. Automatic classification algorithms provide more sophisticated alternatives for false alarm reduction.

The work presented here demonstrates an automatic classification algorithm on a data set collected in a littoral environment. The data set contains a large amount of false alarms, particularly close to the coast, but does not contain any submarine target detections. Synthetic submarine echoes are therefore added to the sonar data set.

Six features are extracted from the hybrid synthetic-recorded data set. The features are fed into supervised machine learning schemes. The performance of each scheme is presented as receiver operating characteristic curves.

I. Introduction

Sea trials in littoral environments have shown that high–resolution, active sonars generate particularly many false alarms in the presence of terrain features, such as seamounts, underwater ridges, and man-made objects, such as ship wrecks and pipelines [4], [9], [15], [16].

Here supervised machine learning algorithms are applied on a sonar data set collected during the New Array Technology 3 (NAT3) programme. NAT3 was a joint research programme between Thales Underwater Systems, TNO, FFI, and the French, Dutch, and Norwegian navies. We have simulated echoes from four different submarine targets and integrated them into the recordings. The acoustic raytrace model, LYBIN [5], is used to estimate the intensity of the received echoes from the simulated submarines.

Standard signal processing following the steps presented in [2] up to cluster level is applied on the hybrid recorded-synthetic data set. The resulting clusters are then assigned classes; either submarine or false alarm. Since the submarine positions are known, this is easily managed.

Various features are extracted directly from the clusters or estimated using the cluster centroid as input, *e. g.* probability of false alarm rate inflation [9]. The performance of each feature is assessed by presenting receiver operating characteristic (ROC) curves [22].

Finally, the features are input into four different machine learning algorithms; *k* Nearest Neighbours [23], Naive Bayes [24], ID3 [18] and Neural networks [8]. The performance of the algorithms are presented and compared through the use of ROC curves.

March 16, 2015

II. Method

A. Synthesising and processing data

A major concern when publishing results from classification algorithms employed on sonar data containing submarines is the confidential nature of such data sets. Although the submarine echo is classified, the remaining sonar data, particularly from experimental systems such as the one considered in this paper, are not necessarily classified. Our solution is to synthesize submarine echoes which are then added to a sonar data set, in which no echoes from submarines were previously present.

The approach employs the acoustic raytrace model LYBIN [5] to estimate the eigenrays [11] between the sonar and the synthetic submarine. Due to two-way propagation, all eigenrays are combined with each other in order to find all echo arrivals from the submarine. The submarine is assumed to be a point target, thus, multiple reflectors are not considered.

Each arrival is characterized by its echo level, EL, arrival angles (horizontal, θ, and vertical, ϕ), and arrival time, t. The vertical arrival angle and arrival time are determined using LYBIN. The horizontal arrival angle is simply estimated from the geometry. The echo level, EL, is estimated as follows:

$$EL = SL - TL_1 - TL_2 + TS + 10\log BT - L, \quad (1)$$

where SL is the sonar source level. TL_1 and TL_2 are the transmission loss to and from the submarine, respectively, which are estimated using LYBIN [5]. TS is the target strength, where the aspect angle dependence is taken into account using the TAP model [10] (50 m long target with 5 m diameter). B and T are the frequency bandwidth and pulse length, respectively, and their product is the assumed gain from the matched filter. L is the assumed coherence loss in the processing, which is here assumed to be 5 dB.

The synthetic submarine echoes are then added to the beamformed and matched filtered sonar data, S, in order to generate the synthetic data $\widehat{S}$:

$$\widehat{S}(j,k) \quad = \quad S(j,k) + \sum_{l} \delta(\tau_k - t_l) b(\alpha_j, \theta_l) 10^{\frac{EL(l)}{10}} \qquad (2)$$

where j is the beam number, k is the time sample number in the sonar data, and l is the submarine echo number. Note that several synthetic submarines may be used, and that each submarine may result in several echoes due to multi-path. τ_k is the arrival time of sample number k. The horizontal beam response, b, depends on the beam steering angle α_j and the horizontal arrival angle θ_l. The horizontal beam response is included in order to synthesise the effect of strong echoes influencing more than one beam.

A cell-averaging constant false alarm rate filter [12] is used to normalize the data (normalization window of 0.5 s and guard band of 0.25 s). Note that the normaliser output, S_N, is considered an estimate of the SNR. A threshold of 12 dB is applied in order to find detections. Finally, clustering, following the steps of Beerens [2], is employed. The template used for the clustering has a length of 250 m and a width of three beams. This template is significantly larger than the one used in [2], because we would also like a cluster to contain all multipath arrivals from the same target, not just multiple reflectors.

B. Features

Six different features are extracted from each cluster:

- Max SNR - The maximum SNR within the cluster, where SNR is the normalised output.
- Mean SNR - The mean SNR in the cluster.
- Std bearing - The standard deviation of the bearing of all samples in the cluster.
- Size - The total amount of samples in the cluster.

All these features are easily extracted from the data processed as described in the previous section. In addition, the probability of false alarm rate inflation (FARI) is included, as described in [9]. FARI is a signal-processing induced phenomenon that occurs when the reverberation in the normalisation windows are non-stationary [20]. The area considered in this work is close to the coast and contains many underwater ridges and seamounts. The occurence of FARI has a high likelihood in such an environment. Probability of FARI is estimated for all samples in each cluster, yielding two more features:

- Maximum FARI - The maximum of the estimated FARI probabilities within the cluster.

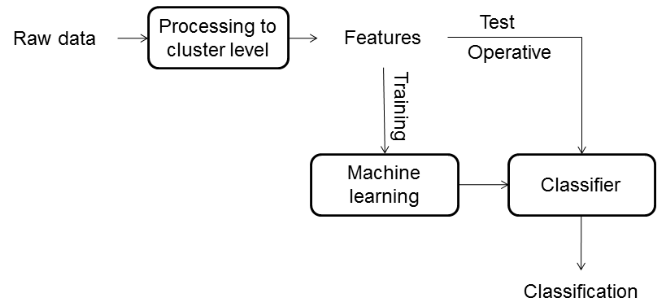

Fig. 1: This flow chart illustrates the supervised learning scheme used to classify the cluster data.

- Mean FARI - The mean of the estimated FARI probabilities within the cluster.

Other features were examined as well, but are not included here either due to poor performance or because the information contents of those features correlated strongly with one of the included features.

C. Machine learning

After extracting the mentioned features from the echoes, these values are fed into a machine learning system in order to generate a classification algorithm. The flow chart in Fig. 1 illustrates the procedure. In order to be able to get a reliable estimate of the performance of the various machine learning algorithms, the data set is split into two parts: One part, the *training set*, consisting of 70% of the echoes, is fed into the machine learning algorithms, while the remaining part, the *test set*, was not presented to the algorithms during training, but was retained for final testing. This allows us to get a reliable estimate of how well the machine learning algorithms can be expected to perform on new, previously unseen, data.

Four different machine learning algorithms were tested. In k *Nearest Neighbours* [23], new instances are classified based on comparing them with the k most similar instances in the training set. The probability of submarine is estimated as $\frac{n_{sub}}{k}$, where k is the number of instances from the training set with which the new instance is compared, and n_{sub} is the number of submarine instances among these instances. We also ran tests where we applied the LaPlace correction, replacing the above fraction with $\frac{n_s ub+1}{k+1}$. Finally, we also tested assigning weights $\frac{1}{d^2}$ to the k nearest instances, where d is the distance from the instance to the instance to be classified, so that the most similar instances had more influence on the outcome than more distant ones.

Somewhat similarly, *Naive Bayes Classification* [24] also bases its classifications on comparisons with instances in the training set. Instead of simple similarity comparisons, it computes probability distributions for the various features and combines these probabilities to compute a probability estimate for the new instance. Since this algorithm requires discrete feature values, we applied a simple discretization algorithm in advance, dividing each feature value into a fixed number of bins in such a way that each bin contained the same number of instances from the training set. Additionally, we tested the algorith both with and without the LaPlace correction, as explained above.

The third algorithm tested, *ID3* [18], builds a decision tree based on the training set in a greedy, top-down manner. We tested the algorithm both with and without the LaPlace correction [17], together with and without Reduced Error Pruning using 30% of the training set as a validation set [19]. Also, we tested with discretization in advance as explained above, in addition to the C4.5 approach of creating thresholds "on the fly" when selecting tests for continuous parameters [19].

Finally, we tested training of *Neural Networks* [8] using Resilient Propagation [21] with varying numbers of hidden neurons, as well as different strategies to avoid overtraining. We tested using Early Stopping [13] (based on a validation set consisting of 30% of the training data), Weight Decay [14] with various λ values, as well as without any overfitting avoidance strategy at all. Additionally, we tested training the neural networks with restarts, that is, repeating the training a number of times, each time assigning a new set of random values to the weights of the network, and finally choosing the network with best performance on the training data.

Additionally, all the above methods were enhanced with *Bagging* [3], a meta-method that re-runs the machine learning algorithm several times, using different random resamplings of the traning set each time. The resulting classifiers are combined into one by averaging their results. In many cases this can improve the performance of the machine learning algorithm significantly, especially for unstable methods like ID3 and training of Neural Networks [1].

Each of the different machine learning algorithms has several parameters that need to be set. We performed a simple manual tuning of these parameters, by running each algorithm on the training set several times with different parameter values, and for each method selecting the set of parameters that yielded the best results on the test set.

For k Nearest Neighbours, the best results were achieved when using $k = 100$, weighting instances by distance, using the LaPlace correction and without using bagging. For ID3, the best results were achieved without using the LaPlace correction, using pruning, no discretization and bagging with 20 resamples. The best classifier using Naive Bayes were found using the LaPlace correction, discretizing with 20 bins and no bagging, while the best results using neural networks were obtained using 20 hidden neurons, no overfitting avoidance strategy, 10 restarts and bagging with 20 resamples.

III. Data

The data set used was recorded during the Clutter Experiment 02 (CEX02), which was carried out in 2002 as a part of the New Array Technology 3 (NAT3) programme. This research programme was a collaboration between Thales Underwater Systems, TNO, FFI, and the French, Dutch, and Norwegian navies. The intent was to test, at sea, new processing methods and algorithms for a low-frequency, towed

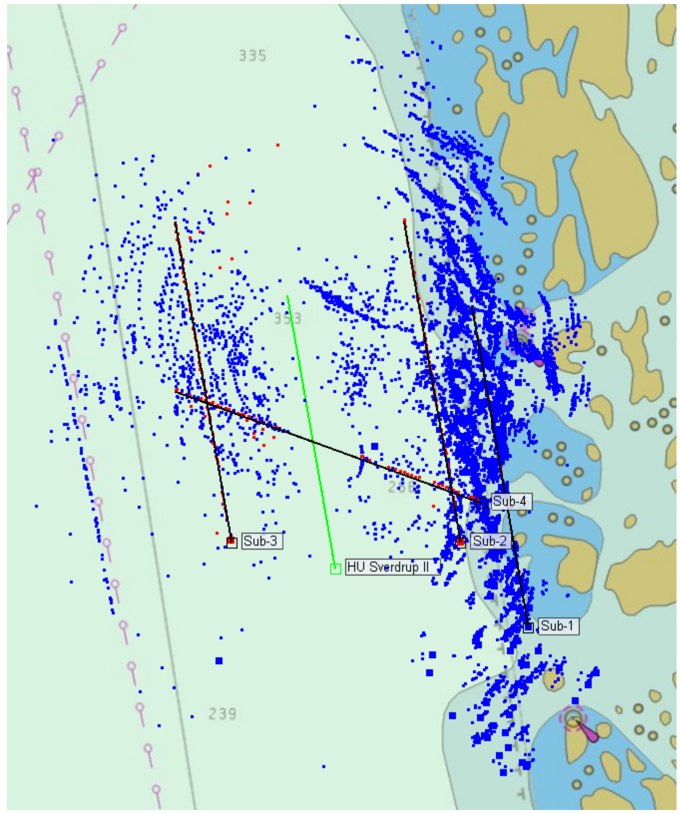

Fig. 2: The paths of the sonar vessel (green) and the four synthetic submarines (yellow). The dots are cluster centroids. The blue dots are classified as false alarms, while the red dots are classified as submarines.

linear array sonar system. CEX02 was designed to maximize the amount of false alarms and was therefore carried out close to the coast in the Norwegian Trench, see Fig. 2.

The data set consists of 80 hyperbolic frequency modulated (HFM) pulse transmissions. The pulse length was 2 s and its frequency bandwidth was 800 Hz. The receiver array consisted of 64 triplet hydrophones [7] spaced at half a wave length.

Four different synthetic submarines were added to the data set, following the steps described in section II-A. The paths of the submarines are plotted in Fig. 2.

IV. Results

The processing described in section II-A was applied on the data set described in section III. Fig. 2 shows the resulting cluster centroids along with sonar and submarine positions for all 80 transmissions.

The six features described in section II-B were extracted for each cluster. Each set of features (from a single cluster) is defined as an *instance*. The instances are divided into two groups; false alarms and true alarms. Instances where the corresponding cluster contains a sample that coincides with a submarine position are assumed to be true alarms. All other instances are assumed to be false alarms.

An efficient way of measuring how well a feature distinguishes a true alarm from a false alarm is to display the receiver operating characteristic (ROC) curves. The ROC curves are estimated by applying a series of threshold values to the features, where each threshold value yields a single *classification rate* and *false alarm rate*. All instances above (or below) the threshold are classified as true alarms, while those below (or above) are classified as false alarms. The percentage of false alarm instances classified as true alarms is the false alarm rate, while the percentage of true alarm instances correctly classified as true alarms is the classification rate. Fig. 3 shows the ROC curves for each feature.

The six features were then input into the four different machine learning algorithms that were briefly described in II-C; k Nearest Neighbours, ID3, Naive Bayes Classifier, and Neural Networks. The resulting ROC curves are shown in Fig. 4. The ID3 algorithm outperforms the remaining three algorithms, the k Nearest Neighbours and Naive Bayes algorithms perform better than simepl SNR thresholding, while the Neural Networks classifiers perform comparable to SNR thresholding.

The results from one of the ID3 classifiers are presented geographically in Fig. 2, where all clusters with a probability of being a submarine above 0.9 are plotted in red, all other clusters are plotted in blue. Notice that there are only a few false alarms, yet most of the clusters on submarine 3 and 4 are classified as submarines. Submarine 1 and 2 both move through a difficult area that is dominated by reverberation from strong upslopes and underwater ridges and mountains. The classification algorithm is able to classify submarine 2 correctly approximately half the time, and submarine 1 occasionally. A reduction of the threshold on the Neural Net output would increase the classification rate, but at the cost of increased false alarm rate.

There are two obvious reasons why the proposed method has difficulties in classifying clusters on submarine 1 and 2 correctly. Firstly the submarines move through an area with strong reverberation, which implies that their SNRs are low. The maximum SNR feature is, according to Fig. 3, the most important feature. Also, the FARI-related methods have limited usefullness in such areas, as they predict a high probability of FARI for all positions close to the coast, see Fig. 5. Submarine 1 is well within the area where a high probability of FARI is predicted, while submarine 2 is in its outskirts.

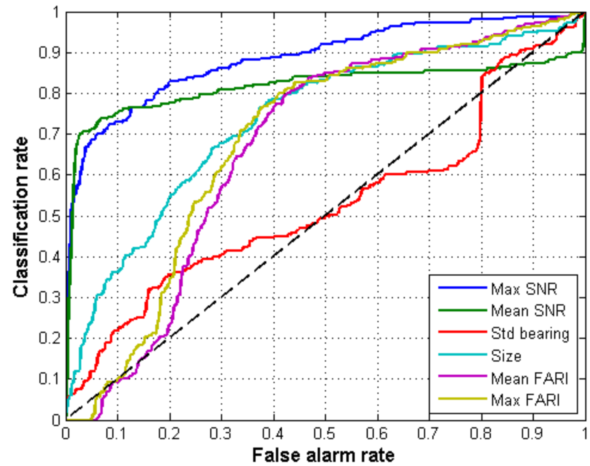

Fig. 3: Receiver operating characteristic curves for each feature.

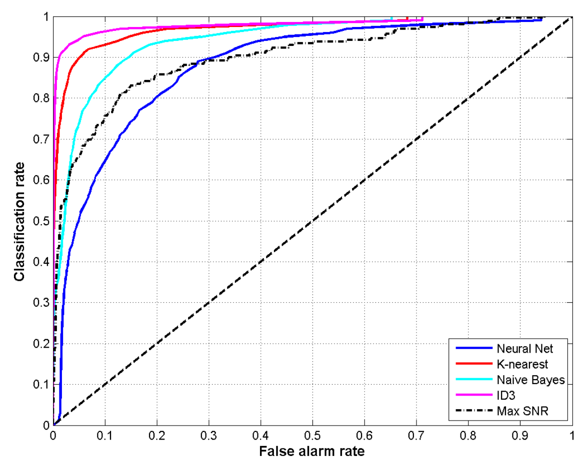

Fig. 4: Receiver operating characteristic curves for each machine learning algorithm. The maximum SNR feature is included for reference.

V. Conclusion

Four different machine learning algorithms have been applied on six features extracted from a hybrid synthetic-recorded data. Three of the algorithms, ID3, k Nearest Neighbours and Naive Bayes classification, are quite successful in classifying the synthetic submarines correctly.

The submarines followed straight paths through both easy areas (flat bottom, low reverberation) and more challenging areas (close to the coast, strongly varying bathymetry, high reverberation). The proposed method achieved a very high classification rate in the easy areas, but had difficulties in classifying submarines moving through a cluttered environment close to the coast. The strong reverberation close to the coast results in low target SNR and high probability of false alarm rate inflation, which strongly reduces the performance of four of the six included features. Improved classification of the submarine in the cluttering environment may be achieved either by decreasing the threshold applied on the Neural Net output, or by including more features that are less sensitive to reverberation. Track level features, such as track kinematics [6], may be useful in such a domain.

Acknowledgment

The authors would like to thank the partners of the NAT III project for making the sonar data from the sea trial available. A particular thanks to TUS and TNO for processing the data

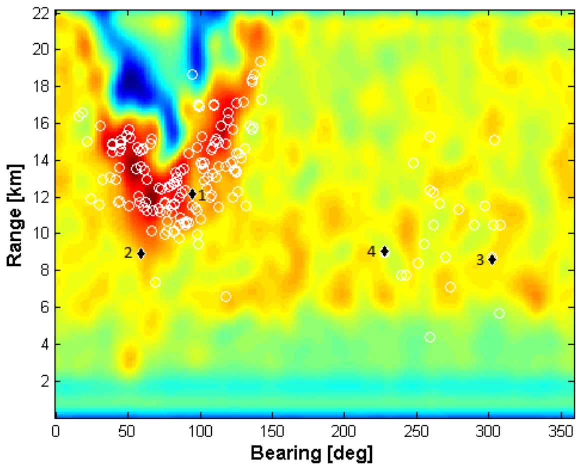

Fig. 5: Probability of FARI as a function of range and bearing (red= high, blue = low) for a single transmission. The white circles are the cluster centroids for the same transmission. The black diamonds incidate the positions of the four submarines.

essential to this work, and to FFI, TNO, RNoN and RNLN for making vessels available for the sea trials.

REFERENCES

[1] Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning **36**(1-2), 105–139 (1999)

[2] Beerens, S., Boek, W.: A robust algorithm for lfas target classification. Undersea Defence Technology-UDT Europe (2007)

[3] Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)

[4] Chotiros, N., Boehme, H., Goldsberry, T., Pitt, S., Lamb, R., Garcia, A., Altenburg, R.: Acoustic backscattering at low grazing angles from the ocean bottom. part ii. statistical characteristics of bottom backscatter at a shallow water site. The Journal of the Acoustical Society of America **77**(3), 975–982 (1985)

[5] Dombestein, E., Wegger, K.E.: Predicting sonar false alarm rate inflation using acoustic modeling and a high-resolution terrain model. Tech. Rep. 2014/00512, FFI (2014)

[6] Grimmett, D., Wakayama, C.: Specsweb post-tracking classification method. In: Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on, pp. 1–8. IEEE (2011)

[7] Groen, J., Beerens, S., Been, R., Doisy, Y., Noutary, E.: Adaptive port-starboard beamforming of triplet sonar arrays. Oceanic Engineering, IEEE Journal of **30**(2), 348–359 (2005)

[8] Haykin, S.: Neural networks: a comprehensive foundation (1999)

[9] Hjelmervik, K.T.: Predicting sonar false alarm rate inflation using acoustic modeling and a high-resolution terrain model. Oceanic Engineering, IEEE Journal of **35**(2), 278–287 (2010)

[10] Hodges, R.P.: Underwater acoustics: Analysis, design and performance of sonar. John Wiley & Sons (2011)

[11] Jensen, F.B.: Computational ocean acoustics. Springer (1994)

[12] Mahafza, B.R.: Radar systems analysis and design using MATLAB. CRC press (2000)

[13] Mitchell, T.M.: Machine learning. wcb (1997)

[14] Moody, J., Hanson, S., Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. Advances in neural information processing systems **4**, 950–957 (1995)

[15] Prior, M.K.: A scatterer map for the malta plateau. Oceanic Engineering, IEEE Journal of **30**(4), 676–690 (2005)

[16] Prior, M.K., Baldacci, A.: The physical causes of clutter and its suppression via sub-band processing. In: OCEANS 2006, pp. 1–6. IEEE (2006)

[17] Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning **52**(3), 199–215 (2003)

[18] Quinlan, J.R.: Induction of decision trees. Machine learning **1**(1), 81–106 (1986)

[19] Quinlan, J.R.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)

[20] Richards, M.A.: Fundamentals of radar signal processing. Tata McGraw-Hill Education (2005)

[21] Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: Neural Networks, 1993., IEEE International Conference on, pp. 586–591. IEEE (1993)

[22] Urick, R.J.: Principles of underwater sound for engineers. Tata McGraw-Hill Education (1967)

[23] Van Der Heijden, F., Duin, R., De Ridder, D., Tax, D.M.: Classification, parameter estimation and state estimation: an engineering approach using MATLAB. John Wiley & Sons (2005)

[24] Zhang, H.: The optimality of naive bayes. AA **1**(2), 3 (2004)