Authentication of Underwater Acoustic Transmissions via Machine Learning Techniques

L. Bragagnolo*, F. Ardizzon*, N. Laurenti*, P. Casari§, R. Diamant‡, S. Tomasin*
*Department of Information Engineering, University of Padova, Italy
§Department of Information Engineering and Computer Science, University of Trento, Italy
‡Department of Marine Technologies, University of Haifa, Israel

***Abstract*—We consider the problem of discriminating a legitimate transmitter from an impersonating attacker in an underwater acoustic network under a physical layer security framework. In particular, we utilize features of the underwater acoustic channel such as the number of taps, the delay spread, and the received power. In the absence of a reliable statistical model of the underwater channel, we turn to a machine learning technique to extract the feature statistics and utilize them to distinguish between legitimate and fake transmissions. Numerical results show how, using only four channel features as input of either a neural network or an autoencoder, we achieve a good trade off between false alarm and detection rates. Moreover, cooperative techniques fusing soft decision statistics from multiple trusted nodes further outperform the discrimination capability of each separate node. Data from a sea trial carried out in Israeli eastern Mediterranean waters demonstrate the applicability of our approach.**

***Index Terms*—Authentication; underwater acoustic channel; physical layer security.**

I. Introduction and related work

Underwater acoustic networks (UWANs) are becoming a feasible option for many oceanic activities that require telemetry, communications, coordination among static and mobile devices, or the periodic monitoring of a given area. With the broadening of the applications that UWANs can support and with the appearance of the first underwater communication standard JANUS [1], however, greater (cyber)security concerns are starting to appear. Key types of attacks that affect UWAN vary from simple signal jamming to impersonation attacks, from attacks to routing protocols to attempts of breaking pre-agreed cryptographic keys used for data exchanges among the nodes [2], [3].

While some signaling and networking protocols may offer a first barrier against attacks on UWANs [4]–[7], a recent trend explores the fundamental characteristics of the underwater acoustic channel (UWAC) to secure underwater communications [8]. Among the first examples of underwater physical layer security, Kulhandjian et al. exploit jamming to disturb unwanted receptions at an eavesdropper, while still allowing communications between a pair of legitimate transceivers [9].

UWACs are known to decorrelate easily in space, and to have a limited time coherence [10], [11]. The statistics of the channel features (e.g., the number of relevant channel taps, the delay spread, and the power of each tap) vary slowly over time [8] and can thus be used to validate a legitimate transmission.

Now, consider an attacker attempting to impersonate a legitimate transmitter in an UWAN, whose trusted nodes collaborate to detect the attack. Trusted nodes can cooperate to distinguish the channel footprint of the legitimate and impersonating nodes, by comparing the statistics of their channel features. The protocol we proposed in an earlier work [8] authenticates packets based on the agreement between the channel statistics across different transmissions. However, given the complexity of the channel statistic, the models obtained by estimation may be mismatched thus affecting the outcome of the process.

In this paper, we resort to machine learning for the robust identification of an impersonating attacker in an UWAN. Our methods include two steps. First, for each trusted node we train a neural network (NN), which outputs a real number representing a soft decision on the authenticity of the received packets. Second, we fuse the NN outputs from all trusted nodes to finally decide on the authenticity of the transmission. On both steps we consider two cases where the dataset of observed features composed of labeled samples under either i) both nominal and attack conditions or ii) only nominal conditions. The latter scenario occurs when the position of the attacker is unknown. In this case, we resort to autoencoder (AE) NNs. Considering that good results are achieved with small networks, we conclude that these are feasible authentication solutions on devices with limited computational power.

We test our scheme both on simulated channels and on data from a sea experiment carried out in the eastern Mediterranean sea near Hadera, Israel. Our results confirm that our proposed scheme successfully distinguishes between authentic and impersonating transmissions, without the complexity and locality of other schemes available in the literature such as [8]. In particular, in our simulations we successfully discriminate between the legitimate transmitter and the attacker even when they are located close to each other, albeit at different depths. Results for the sea experiment support the same conclusion in a realistic environment. Here, we show that a few hundreds meters between the legitimate transmitter and the attacker are sufficient to tell the two nodes apart, even when relying on a single trusted receiver.

The remainder of this paper is organized as follows. In Section II, we describe our system model. We proceed with our authentication protocol in Section III and provide both simulation and experimental results in Sections IV and V, respectively. Finally, we draw concluding remarks in Section VI.

II. System Model

We consider an UWAN composed of a legitimate transmitter, namely Alice, a set of N trusted receivers $\mathcal{B} = \{B_n, n = 1, \ldots, N\}$, and an attacker, namely Eve. In the following, we will refer to any trusted receiver as Bob. We assume that all nodes are loosely synchronized, and that each packet has a unique sequential identification number (ID). This allows different receivers to perform a distributed cooperative check by observing the same broadcast packet. The exact location of the different receiver nodes is unknown to both the trusted receivers and Eve.

We assume that all trusted receivers are connected to a sink node via a limited-rate, authenticated, and integrity-protected channel, over which they can share their observations. Then, the sink makes the final decision on the authenticity of the received packets. We make a first decision at each node to avoid transmitting each single observation to the sink. This reduces the communication overhead. We model each point-to-point UWAC as a tapped delay line, having power-delay profile $H'_n(t, \tau)$ at time t.

The attacker Eve is a single malicious node. However, our scheme can be straightforwardly extended multiple attackers. We also make no assumption on the contents of the packets, i.e., we assume that the packets sent by Alice and Eve are indistinguishable at the data level.

A. Features for Authentication on UWACs

To assess the authenticity of the received packet, we rely on four channel statistics. Let $x_{i,n}(t)$ be the value of the ith feature with $i = 1, \ldots, 4$ measured at time t by node B_n. To extract the features, we zero out low-power arrivals in the power-delay profile, i.e.,

$$H_n(t, \tau) = \begin{cases} 0 & |H'_n(t, \tau)| < T_h, \\ H'_n(t, \tau) & |H'_n(t, \tau)| \geq T_h, \end{cases} \tag{1}$$

where we choose T_h to obtain a desired false alarm probability, as detailed in [12]. Call $\mathcal{S}_n(t)$ the set of delays of all channel arrivals that remain after thresholding. We consider the following four features:

1–Number of channel taps. The estimated number of relevant taps revealing the spread of the acoustic channel:

$$x_{1,n}(t) = |\mathcal{S}_n(t)| \, . \tag{2}$$

2–Average tap power. The average power of the relevant taps, which reflects how diverse and sparse the channel is:

$$x_{2,n}(t) = \frac{1}{|\mathcal{S}_n(t)|} \sum_{\tau \in \mathcal{S}_n(t)} |H_n(t, \tau)| \, . \tag{3}$$

3–Relative root mean square (RMS) delay. This feature reflects the delay spread of the channel. Let $\tau_0 = \min\{\tau : \tau \in \mathcal{S}_n(t)\}$ be the delay of the first tap, then the relative RMS delay is

$$x_{3,n}(t) = \left(\frac{1}{|\mathcal{S}_n(t)| - 1} \sum_{\tau \in \mathcal{S}_n(t), \tau \neq \tau_0} (\tau - \tau_0)^2 \right)^{1/2} . \tag{4}$$

4–Smoothed received power. This feature accounts for the overall attenuation in the channel. To track the variation of power over time, let $q_{n,t}$ be the power of a symbol received by node n at time instance t. Given a user-defined parameter $0 \leq \alpha \leq 1$, we recursively compute the smoothed received power as

$$x_{4,n}(t) = \alpha \, q_{n,t} + (1 - \alpha) \, x_{4,n}(t') \, , \tag{5}$$

where $x_{4,n}(t')$ is the smoothed received power of the previous symbol received at time t'.

We choose these features, because their statistics are stable over time and depend strongly on the transmitter's location [8]. We use the estimated statistics for authentication purposes: therefore, by comparing the channel's features, Bob can distinguish between packets arriving from sources located at different locations.

III. Authentication Protocol

In the absence of a reliable statistical model of the UWAC, we turn to a NN to distinguish between a legitimate transmission and a fake one. We consider two alternative scenarios, depending on the data available to train the NN. In the first scenario, Bob has observations available from both Alice and Eve; instead, in the second scenario, Bob has observations only from Alice, as would be the case if the statistics of Eve's channel features are unknown. Therefore, in the first scenario the NN operates as a two-class classifier, whereas it operates as a one-class classifier (or AE) in the second scenario.

Considering the received packet ϕ, we formulate the authentication problem as a binary hypothesis test, where the two hypotheses are:

- $\mathcal{H}_0$ (legitimate): ϕ was transmitted by Alice and,
- $\mathcal{H}_1$ (non legitimate): ϕ was not transmitted by Alice.

Calling $\mathcal{H} \in \{\mathcal{H}_0, \mathcal{H}_1\}$ the true class of the received packet, the decision Bob makes based on $\mathcal{H}$ is $\hat{\mathcal{H}} \in \{\hat{\mathcal{H}}_0, \hat{\mathcal{H}}_1\}$. We account for two cases of misclassification, i.e., false alarms (FAs), where the transmitter considers a signal transmitted by Alice as fake, and missed detections (MDs), where Bob considers a signal from Eve as legitimate. The FA probability is defined as $p_{\text{FA}} = \mathbb{P}(\hat{\mathcal{H}} = \mathcal{H}_1 | \mathcal{H} = \mathcal{H}_0)$ and the MD probability is defined as $p_{\text{MD}} = \mathbb{P}(\hat{\mathcal{H}} = \mathcal{H}_0 | \mathcal{H} = \mathcal{H}_1)$.

A. Local NN-Based Authentication

The aim of a NN is to provide a test function $f(\boldsymbol{x})$. In our case, $f(\boldsymbol{x}) = -1$ and $f(\boldsymbol{x}) = 1$ ideally when $\mathcal{H} = \mathcal{H}_0$ and $\mathcal{H} = \mathcal{H}_1$, respectively. We model the NN as a function $\mathbb{R}^4 \longrightarrow \mathbb{R}$ composed of Q stages, typically called *layers*: we call layer 0 the *input layer*, the last stage the *output layer* and the remaining layers *hidden layers*. The first layer input is mapped to a vector $\boldsymbol{y}^{(0)}$, whereas the output of the output layer is $y^{(Q)}$. We represent the output of the kth neuron of the qth layer as $y_k^{(q+1)} = \psi^{(q)}\big(\boldsymbol{w}_k^{(q)} \boldsymbol{y}^{(q)} + b_k^{(q)}\big)$, where $\psi^{(q)}(\cdot)$ is the neuron *activation function*, $\boldsymbol{y}^{(q)}$ is the output of the previous layer q, $b_k^{(q)}$ is a bias value and $\boldsymbol{w}_n^{(q)}$ is a vector of weights. We consider only feedforward NNs [13] with no loops between the layers. While the activation functions are decided a priori,

weights $\boldsymbol{w}_k^{(q)}$ are optimized during the algorithm's learning phase. Finally, considering the (single-node) output of the last layer $y_1^{(Q)}$, we choose function f as

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } y_1^{(Q)} \geq \lambda \\ -1 & \text{if } y_1^{(Q)} < \lambda \end{cases} \tag{6}$$

where λ is chosen a priori depending on a target p_{FA} value. Notice that by increasing λ we reduce p_{FA} and increase p_{MD}; vice-versa, decreasing λ reduces p_{MD} while increasing p_{FA}.

B. Local Autoencoder-based Authentication

An AE is an unsupervised NN trained to replicate its input at the output [13], [14]. An AE is composed of an encoder, a hidden layer with $M < N$ nodes, and a decoder. The task of the encoder, composed of Q_{e} layers, is to project the input vector $\boldsymbol{x}$ into a lower dimensional space of size M. The task of the decoder is to retrieve the original input vector from the encoded word. Notice that the reconstruction process is not perfect, and depends on the size of the training dataset and of the hidden layer. While a larger hidden layer eases the reconstruction of the input, a smaller hidden layer enables a more accurate characterization of the training set.

Because we use the AE for one-class classification, our aim is not only to properly reconstruct input features from a legitimate transmission, but also to yield a significant reconstruction error when the input is a set of features from a transmission by Eve. If this occurs, we obtain an authentication test by comparing how good is the match between the input and the output of the AE. To this end, by using the smallest size of the hidden layer that properly reconstructs the legitimate input of the training set, we ensure that an input with different characteristics, i.e., coming from an impersonation attack, is not properly reconstructed. Still, note that the training phase is performed using only channel features from legitimate transmissions.

Formally, let $\boldsymbol{y}^{(Q)}(\boldsymbol{x}) = [y_1^{(Q)}(\boldsymbol{x}), \ldots, y_4^{(Q)}(\boldsymbol{x})]$ be the output of the trained AE: we associate the following mean-square error (MSE) function to the input feature vector $\boldsymbol{x}$

$$\Gamma(\boldsymbol{x}) = \frac{1}{4} \sum_{i=1}^{4} |x_i - y_i^{(Q)}(\boldsymbol{x})|^2, \tag{7}$$

which provides the test function

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Gamma(\boldsymbol{x}) < \lambda, \\ -1 & \text{if } \Gamma(\boldsymbol{x}) \geq \lambda, \end{cases} \tag{8}$$

where λ depends on the target false alarm probability, p_{FA}.

C. Neyman-Pearson Test

For comparison purposes, we consider also the Neyman-Pearson (N-P) test. Let $p(\boldsymbol{x}|\mathcal{H}_i)$ be the probability density function (PDF) of observation $\boldsymbol{x}$ given that $\phi \in \mathcal{H}_i$. We compute the log likelihood ratio (LLR) as

$$\mathcal{M}(\boldsymbol{x}) = \ln \frac{p(\boldsymbol{x}|\mathcal{H}_0)}{p(\boldsymbol{x}|\mathcal{H}_1)}, \tag{9}$$

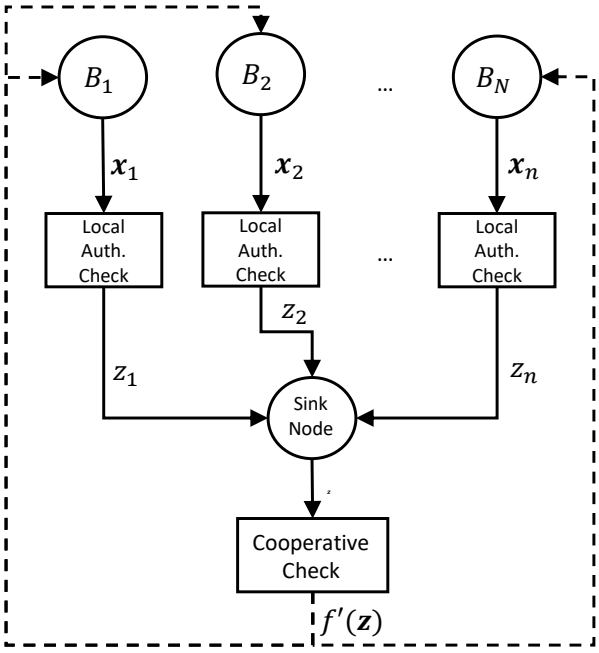

Fig. 1. Scheme of the two-step cooperative authentication protocol.

and compare it to a threshold λ to obtain the N-P test

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \mathcal{M}(\boldsymbol{x}) > \lambda, \\ -1 & \text{if } \mathcal{M}(\boldsymbol{x}) \leq \lambda. \end{cases} \tag{10}$$

However, as stated in Section II, we do not have a general statistical model for the UWAC, hence we cannot analytically derive $p(\boldsymbol{x}|\mathcal{H}_i)$ as needed for the N-P test. However, to compare the performance of our solutions, we will infer $p(\boldsymbol{x}|\mathcal{H}_i)$ by estimating it directly from our data set.

D. Cooperative Authentication

We consider the two-step authentication protocol of Fig. 1, where each node B_n runs a single-node authentication protocol and transmits to the sink node either $z_n = y_1^{(Q)}$, if B_n uses a NN, or $z_n = \Gamma(\boldsymbol{x})$, uses an AE. To perform the cooperative authentication check, the sink node fuses the information coming from the nodes. We distinguish two cases, depending on the local authentication. In particular,

1) for a NN-based local authentication, the sink can train a second NN for the global decision, because we also have access to the features of transmissions by Eve;
2) for an AE-based local authentication if only data coming from Alice are available for training, the sink node trains another AE.

Moreover, we also propose an additional, simpler approach that can be used in both the two above cases: the sink node can linearly combine the information from the nodes as follows

$$g(\boldsymbol{z}) = \frac{1}{N} \sum_{n} z_n \,. \tag{11}$$

This also helps counter the fact that solution 2) tends to perform poorly, as will be clearer from simulation results.

Denote in general the output of the cooperative check as $g(\boldsymbol{z})$. As for local authentication, we compare $g(\boldsymbol{z})$ with a threshold λ' such that

$$f'(\boldsymbol{z}) = \begin{cases} 1 & \text{if } g(\boldsymbol{z}) < \lambda', \\ -1 & \text{if } g(\boldsymbol{z}) \geq \lambda'. \end{cases} \tag{12}$$

The result of this test is then broadcast by the sink to all the nodes.

IV. Simulation Results

We consider the channel model of [15]: in particular we model our UWAN with

- 3 legitimate receivers, namely B_1, B_2, and B_3, placed at different depths;
- Alice, located at a depth of 20 m;
- Eve, located close to Alice at depth of 480 m.

Notice that we challenge our solution by placing Eve and Alice close to each other; if Alice and Eve are located farther away, it may be easier for a node to check the authenticity of the packets, e.g., by thresholding the received power. Fig. 2 shows an example of the simulated scenario: for simplicity we did not include the sink node in the figure, assuming that any node may act as the sink node. To generate the data set, we simulate the communication in the UWAC using the Bellhop simulator and the Acoustic Toolbox [15], [16]. In particular, for each node, we pick 500 different position uniformly at random within a sphere of radius 10 m centered on each node's nominal location. For each pair of nodes, this yields 500×500 transmitter-receiver pairs. We ran Bellhop for each pair and obtained a realization of the UWAC. We considered a rough sea surface and modeled the sea bottom with hills of sinusoidal shape, with diameter of 200 m and maximum height of 10 m. To model the sound speed profiles (SSP) we considered the measurements available at [17].

A. Results For the Local Authentication

We start considering the local authentication at each node, where receiver B_1 acts as Bob. As stated in Section III, we compare the proposed approach with the N-P test: however, we cannot analytically compute the PDFs $p(\boldsymbol{x}|\mathcal{H}_i)$. Instead, we estimate them from the channel realizations via kernel density estimation (KDE) [18]. Implicitly, as in [8], we also assume that the features are independent.

The NN has

- 4 nodes on the input layer with the ReLu as activation function,
- 2 hidden layers composed of 3 nodes each with the ReLu activation function,
- one node on the output layer with the sigmoid activation function.

The AE is composed of

- 4 nodes for the input layer, i.e., the encoder, with the ReLu,

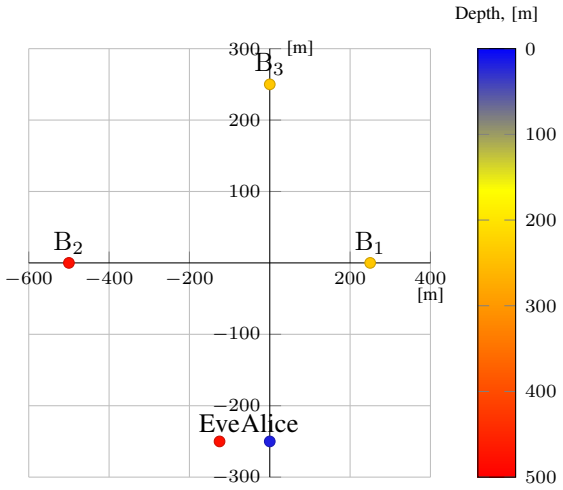

Fig. 2. Example of the simulated scenario.

- a single hidden layer with $M = 3$ nodes, with the ReLu activation function,
- 4 nodes for the output layer, i.e., the decoder with a linear activation function.

For both NN and AE we used 60% of the data set for training, 15% for validation, and 25% for testing.

In our context, conventional machine learning metrics such accuracy are not particularly relevant, since we can directly evaluate the impact of machine learning approaches (including the effect of the training set) on metrics defined for our specific authentication application. Specifically, we focus on the detection error tradeoff (DET) curves, i.e., the value of FA and MD probabilities, for different values of the threshold λ. Fig. 3 shows the obtained DET curve: we observe that, even if Alice and Eve are close to each other, both NN and AE achieve good results. We also observe that the NN outperforms both the AE and the N-P test. In fact, different from the AE, the NN is trained using also data from Eve UWAC realizations. Moreover, we use estimated PDFs in the N-P test and their mismatch with respect to the features' true statistics negatively affects the test performance.

B. Cooperative UWAC Authentication

In this section, we report results using the cooperative authentication strategies described in Section III-D. Fig. 4 shows the DET at the sink node considering that all the nodes have access to both Alice and Eve realizations for training, and use a NN for the single node authentication check. Fusion at the sink is performed with either

1) a global NN with a design analogous to the local NN, but having 3 nodes on input layer and a single 2-node hidden layer, or
2) the linear combination of the local NNs outputs (Eq. (11)).

We observe that both cooperative checks outperform the local authentication. In particular, the NN makes the data

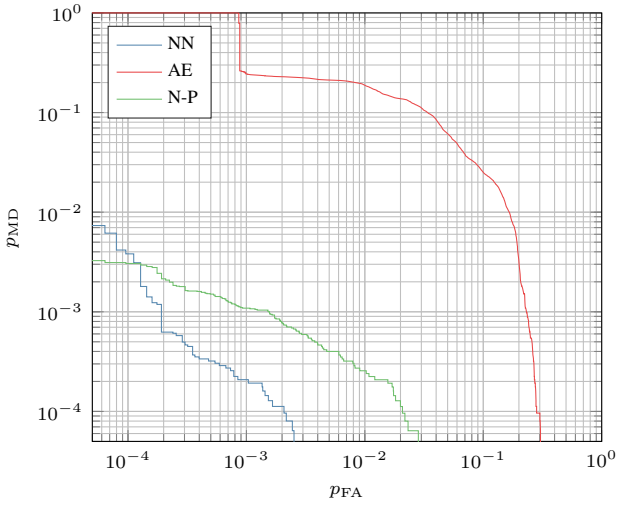

Fig. 3. DET curves for the simulated scenario, using NN, AE, and N-P test.

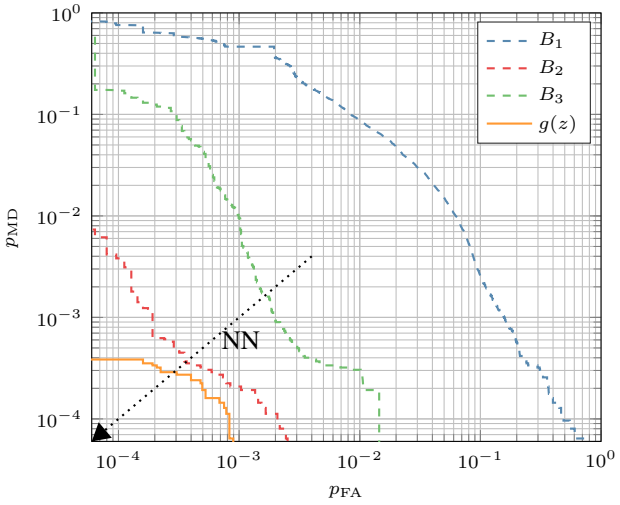

Fig. 4. DET curves for the cooperative authentication check in the simulated scenario. Here, we fuse the results of the NN single node checks.

set separable, i.e., there exists a value of λ' that provides $p_{\mathrm{fa}} = p_{\mathrm{md}} = 0$, thus no line is reported in the log-scale DET figure. Note also that the results of the faster and simpler to train support vector machines (SVM) do not lag much behind the NN, with $p_{\mathrm{fa}} = p_{\mathrm{md}} \approx 10^{-4}$, where the used data set included 250 000 realizations.

Fig. 5 shows the DET obtained using observations from Alice only, i.e., an AE is used each node and the sink node fuses the MSE obtained from the nodes by either

1) a second AE with the same design as the first one but one less node on both input and hidden layer, or
2) the linear combination (11).

While the first solution does not improve the results, the second is shown to be effective. Specifically, it provides lowers probabilities of misdetection and false alarm than those of node in the best position, taking advantage of the less reliable nodes.

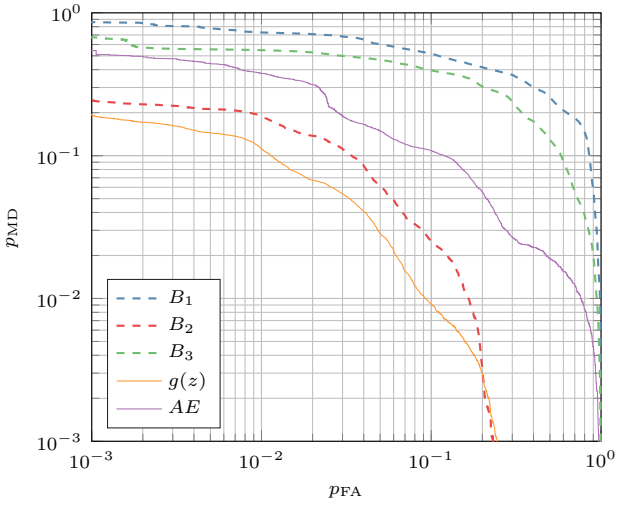

Fig. 5. DET curves for the cooperative authentication check in the simulated scenario. Here, we fuse the local soft decisions of the AEs.

V. Experimental Results

To demonstrate the performance of our authentication protocol in a realistic environment, we repeated the training and evaluation process of the NN also for a data set obtained from a sea experiment. The experiment was conducted near the Hadera coal pier in Israel in May 2017, with the setup shown in Fig. 6. In details, we used

- two projectors, Tx1 and Tx2, acting as Alice and Eve, respectively. Tx1 was deployed from the pier, while Tx2 was deployed from a boat. The distance between Tx1 and Tx2 was roughly 1 km;
- three receivers, Rx1, Rx2, and Rx3. Rx1 and Rx3 were deployed from two floating buoys, while Rx2 was deployed from a boat. The distance between each receiver was approximately 500 m.

Tx1 and Tx2 mounted EvoLogics software-defined S2CR 7/17 modems, and transmitted packets composed of 100 chirp symbols of duration 10 ms in the 7-17 kHz band. The source level was roughly 175 dB re 1 μPa@1m. The receivers used a Cetacean CR1 hydrophone and continuously recorded the raw acoustic data. The data set collects the measurements acquired by Rx2. To process the experiment data, we used the same NN and AE design used for the simulated UWAC.

Using the NN, for the experiments we always have

$$\begin{cases} y_1^{(\mathrm{Q})} > 0.9 & \text{if } \phi \in \mathcal{H}_0, \\ y_1^{(\mathrm{Q})} < 0.9 & \text{if } \phi \in \mathcal{H}_1, \end{cases}$$

Since the two distributions are separable by a threshold, it is possible to find values of λ (e.g., $\lambda = 0.9$) such that $p_{\mathrm{FA}} = p_{\mathrm{MD}} = 0$.

Equivalently, using instead the AE we get

$$\begin{cases} \Gamma(\boldsymbol{x}) < 0.1 & \text{if } \phi \in \mathcal{H}_0, \\ \Gamma(\boldsymbol{x}) > 0.1 & \text{if } \phi \in \mathcal{H}_1. \end{cases}$$

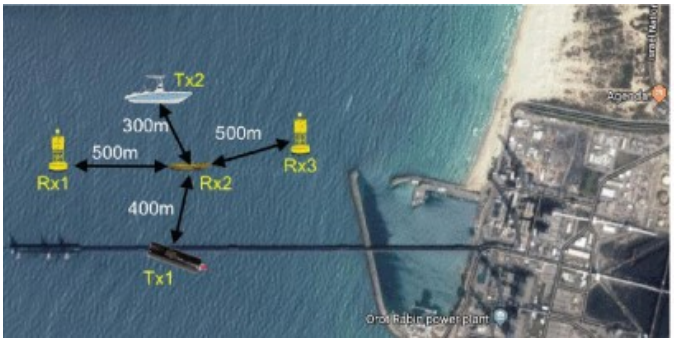

Fig. 6. Setup of the sea experiment in Hadera, Israel.

hence also these distributions are separable after the AE so the Bob is able to perfectly distinguish between Alice and Eve packets.

VI. Conclusions

In this paper, we proposed a novel authentication protocol for underwater acoustic networks that exploits machine learning techniques to assess the authenticity of a received packet. We considered a two-step approach. First, each node authenticates the received packet using a locally-trained NN. Then, each node transmits the (soft) output to a sink node, which fuses the local outputs to make a global decision. To train our networks, we considered a scenario where all nodes train their models with realizations drawn both from legitimate and attacker channels, and a scenario where only a data set of legitimate features was available. While experimental results show that even single-node authentication is effective, simulation results prove that adding the cooperation step makes our protocol able to distinguish between legitimate and attacker packets even when Alice and Eve are relatively close to each other.

Acknowledgment

This work was sponsored in part by the NATO Science for Peace and Security Programme under grant no. G5884 (SAFE-UComm), and by MIUR (Italian Ministry of Education) under the initiative *Departments of Excellence* (Law 232/2016).

References

[1] J. Potter *et al.*, "The JANUS underwater communications standard," in *Proc. UComms*, Sestri Levante, Italy, Sep. 2014.

[2] C. Lal *et al.*, "Secure underwater acoustic networks: Current and future research directions," in *Proc. IEEE UComms*, Lerici, Italy, Aug. 2016.

[3] G. Yang *et al.*, "Challenges and security issues in underwater wireless sensor networks," *Procedia Computer Science*, vol. 147, pp. 210–216, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919302443

[4] G. Dini and A. Lo Duca, "A secure communication suite for underwater acoustic sensor networks," *MDPI Sensors*, vol. 12, pp. 15 133–15 158, 2012. [Online]. Available: http://dx.doi.org/10.3390/s121115133

[5] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, 2019.

[6] E. Souza *et al.*, "End-to-end authentication in under-water sensor networks," in *Proc. IEEE ISCC 2013*, 2013, pp. 000 299–000 304.

[7] F. Campagnaro *et al.*, "Replay-attack countermeasures for underwater acoustic networks," in *Proc. MTS/IEEE OCEANS*, 2020, pp. 1–9.

[8] R. Diamant *et al.*, "Cooperative authentication in underwater acoustic sensor networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 954–968, 2019.

[9] H. Kulhandjian *et al.*, "Securing underwater acoustic communications through analog network coding," in *Proc. IEEE SECON*, Singapore, Jun. 2014.

[10] P. A. van Walree, "Propagation and scattering effects in underwater acoustic communication channels," *IEEE J. Ocean. Eng.*, vol. 38, no. 4, pp. 614–631, 2013.

[11] D. Grimmett and R. Plate, "Temporal and doppler coherence limits for the underwater acoustic channel during the lcas'15 high duty cycle sonar experiment," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–9.

[12] R. Diamant, "Closed form analysis of the normalized matched filter with a test case for detection of underwater acoustic signals," *IEEE Access*, vol. 4, pp. 8225–8235, 2016.

[13] I. Goodfellow *et al.*, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] N. Morozs *et al.*, "Channel modeling for underwater acoustic network simulation," *IEEE Access*, vol. 8, pp. 136 151–136 175, 2020.

[16] M. Porter *et al.*, "Bellhop gaussian beam/finite element beam code," http://oalib.hlsresearch.com/Rays/index.html. Last accessed: June. 2021.

[17] B. Dushaw, "Worldwide Sound Speed, Temperature, Salinity, and Buoyancy from the NOAA World Ocean Atlas," http://staff.washington.edu/dushaw/WOA/. Last accessed: June. 2021.

[18] G. Roussas, *Nonparametric Functional Estimatoin and Related topics*. Springer, 1991, NATO ASI Series.

[19] W. Aman *et al.*, "On the effective capacity of an underwater acoustic channel under impersonation attack," in *Proc. IEEE ICC 2020*, 2020, pp. 1–7.

[20] L. Xiao *et al.*, "Learning-based phy-layer authentication for underwater sensor networks," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 60–63, 2019.

[21] W. Wilson, "Speed of sound in sea water pressure, and salinity," *J. Acoust. Soc. Am.*, no. 6, pp. 641–644, 6 1960.

[22] W. Wang *et al.*, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proc. IEEE CVPR Workshops*, 2014, pp. 496–503.

[23] L. Su *et al.*, "Detecting multiple changes from multi-temporal images by using stacked denosing autoencoder based change vector analysis," in *Proc. IEEE IJCNN 2016*, 2016, pp. 1269–1276.

[24] G. Han *et al.*, "Secure communication for underwater acoustic sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 54–60, Aug. 2015.