On the Quickest Detection Problem for Authentication in Underwater Acoustic Channels

Laura Cardillo[1], Francesco Ardizzon[1], and Stefano Tomasin[1,2]

[1]Department of Information Engineering, University of Padua
[2]Department of Mathematics, University of Padua, Italy, and National Inter-University Consortium for Telecommunications (CNIT), Italy

Francesco Ardizzon, Department of Information Engineering, University of Padua, Via G. Gradenigo 6/B, 35131, Padova, Italy.
email: francesco.ardizzon@phd.unipd.it

Abstract: Underwater acoustic communications are becoming increasingly popular, due to their use in both military and civil applications. Indeed, many of these services require the communications to be authenticated, i.e., the receiver must detect if the received message was transmitted by an impersonating attacker. Several security solutions have been proposed to provide authentication for underwater acoustic channels (UWACs). These typically test a security metric and are based on either cryptography or physical-layer security (PLS), i.e., evaluating the statistical properties of the channel itself.

In this paper, we instigate the problem of quickest detection for physical-layer authentication (PLA) in UWACs, assuming that from a time instant all the signals are sent by the attacker and the defender aims at detecting when an attack starts with the minimum possible delay, by examining the characteristics of the UWAC over which the signal is received. We propose a solution based on recurrent neural networks (RNNs). These neural networks (NNs) process the input taking into account also its past realizations, and thus are suitable to detect anomalous changes that can be tied to an impersonation attack. The proposed approach quickly detects when the attacker is starting its attack activity. Performance are evaluated using an authentication test on an experimental dataset, comparing the performance to the optimal cumulative sum (CUSUM) test which however requires the prior knowledge of the UWAC statistics.

Keywords: *Quickest Detection, Physical Layer Security, Authentication.*

1. INTRODUCTION

Underwater acoustic communications are becoming increasingly popular, due to their use in both military and civil applications. These applications are typically paired with security services, such as a) encryption, allowing a transmitter and receiver pair, namely Alice and Bob, to communicate without disclosing information to any third-party eavesdropper (Eve), or b) authentication, where the receiver aims to assess if the signal and the message has been receiver as intended by the legitimate user, i.e., assuring that the signals were not transmitted by a malicious Eve impersonating Alice or c) integrity protection, ensuring that Eve did not tamper with the signals or their message content.

An increasingly popular solution to provide security is to resort to physical-layer security (PLS) instead of cryptographic solutions. PLS techniques rely on the statistical properties of the channel to provide security. Focusing on physical-layer authentication (PLA), Bob verifies the authenticity of the packet by evaluating the channels. Assuming that legitimate and attacker channels are different, for instance, due to the different transmitters' positions (channel-based PLA). A possible alternative is to look for the non-idealities induced by the transmitters themselves as a fingerprint on the signal (source-based PLA). In this paper, we focus on the former option. A more theoretical introduction to PLS for wireless channels is reported in [1].

The common aspect of PLS authentication is that they typically involve a test, evaluating whether a set of features extracted from the channel impulse (or frequency) response of the received signal matches a predefined statistic, possibly considering also the attacker statistics (if known). Moreover, since in practice a dataset of trusted measurements may be available instead of the prior distribution, a common practice is to resort to machine learning techniques such as neural networks (NNs) or autoencoders, considering the authentication as a classification problem in hypothesis testing context [2–5].

We consider an authentication scenario where the attacker starts transmitting fake signals to the receiver. Thus, while previously Bob was receiving only signals from Alice, from that instant on, Bob will receive instead only signals from Eve. This is actually representative of many security scenarios since, e.g., when a breach of the system is found, the attacker repeats the attack to maximize its chance of success, before the vulnerability is fixed. Bob's aim is then to detect, in the least possible time, the start of the attack, by finding the trade-off between the false detection probability, i.e., the probability of labeling a legitimate signal as fake, and the detection delay, the time between the actual attack start and the detected change. Indeed, choosing a test that is sensitive to sudden changes in the variation may reduce the detection delay, but it also increases the false alarms.

Notice that, in principle, it is still possible to use the typical PLA techniques in a *single-shot* manner, verifying at each instant whether the signal is legitimate or not. However, these tests are memoryless and do not exploit the persistent behavior of the attacks. Hence, we resort to *sequential testing* and frame it as a quick-detection problem. In this context, the cumulative sum (CUSUM) algorithm [6] which relies on the likelihood ratio test (LRT), has been proven to be optimal when i) the collected samples are i.i.d. and ii) the prior distributions are both known [7]. Quickest detection has been applied in various contexts, for instance, to detect cyber-attacks in smart grids [8, 9] or wireless networks [10], or epidemics monitoring [11]. For a survey on quickest detection and its application, the interested reader may check [12].

In this paper, we instigate the problem of quickest detection for PLA in underwater acoustic channels (UWACs). Differently, for instance, from [8, 9] the distribution of neither the legitimate nor the under-attack sample distribution is known a priori. Moreover, in many practical underwater communication scenarios, UWACs are notoriously hard to model since channel pa-

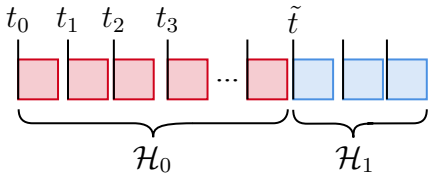


Figure 1: Example of signal reception at Bob over time.

rameters are strongly dependent on the scenario conditions, which may be hard to precisely monitor. For instance, the sound speed profile depends on temperature, depth, and water salinity.

We propose a solution based on recurrent neural networks (RNNs): these NNs process the input signal while tracking also its temporal evolution. In particular, being provided with memory, these are good candidates for solving the quickest detection of anomalous trends on the observed sample statistic that suggest that an attack has started. Performance are evaluated using an authentication test on an experimental dataset. Finally, our solution is then compared to both NNs-based solution and the statistical-based CUSUM test.

The rest of the paper is organized as follows. Section 2 introduces the system model. Section 3 details the proposed protocol. Next, in Section 4 we present the numerical results. Finally, Section 5 draws the conclusion.

2. SYSTEM MODEL

We consider a legitimate agent, Alice transmitting signals to agent, Bob, via a UWAC. Upon packet reception at time t_n, Bob computes from the estimated UWAC response a vector, $\boldsymbol{x}_n$ containing M features. In particular, we extract from the UWAC the number of taps, the average tap power, root mean square (RMS) delay, and smoothed received power. More details about the derivation of these features and the motivation behind this choice of features are reported in [2, 3]. Still, the proposed algorithm does not depend on the particular set of chosen features, and another set can be considered, for instance, those of [13].

Next, at any instant $\tilde{t} > t_0$, Eve starts sending packets instead of Alice. Hence, Bob needs to detect, in the least possible delay, when the attack starts. We remark that we assumed in this paper that after $\tilde{t}$, only Eve packets are received by Bob. We left the analysis of the more general scenario where both Alice and Eve packets are received as future work.

We assume the features collected from the legitimate and the attacker channel to have a different probability density function (PDF), thus it is possible to distinguish between the two by using PLA techniques. In particular, the attacker does not know the channel in advance, thus she cannot precompensate the signal to make it indistinguishable from Alice one at Bob's front-end. The scenario is sketched, from Bob's perspective, in Fig. 1.

We consider a static scenario, where the measurements collected by Bob are time-invariant. Still, when considering dynamic scenarios, for instance considering a scenario slowly changing due to the environmental conditions, we can update the pre-trained old model in a continual learning fashion. We assume that a limited dataset containing both Alice-Bob and Eve-Bob feature vectors is available, which will be later used to train Bob's detector. This could be either collected from the channel itself, assuming that for a limited period of time a second authentication mechanism is in place, or by storing simulated data obtained for different transmitter positions.

3. PROPOSED PROTOCOL

We frame the authentication problem as a binary hypothesis testing, where the two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ correspond to the legitimate and under-attack case, respectively. To detect whether a single observation belongs to $\mathcal{H}_0$ or $\mathcal{H}_1$ the optimal statistical solution would be to resort to the LRT.

In this case, however, we can exploit the stationarity of the attacks, i.e., we can use the sequence of (past) measurements. While optimal for the single shot case, the LRT does not take advantage of the past evolution of the observation. Thus, we need to consider sequential testing, where the detector is provided with memory.

First, we formalize the problem. For a test variable v_n which is a function of both the current and the past inputs, we define the *detected change time* $t^\star \triangleq \min\{t_n | v_n > \lambda\}$, where λ is a threshold chosen by the user to meet a predefined false alarm probability. Following Pollack formulation [14], our aim is then to minimize

$$\min_{\tilde{t}} \sup_{t^\star \geq 1} \mathbb{E}_v[\tilde{t} - t^\star \mid \tilde{t} \geq t^\star] \,, \quad \text{subj. to } \mathbb{E}_\infty[\tau] \geq \beta \,, \tag{1}$$

thus designing the test that detects the change, i.e., the start of Eve's transmission, with the lowest possible delay.

For time-invariant processes where the PDFs $p(\boldsymbol{x}_n|\mathcal{H}_0)$ and $p(\boldsymbol{x}_n|\mathcal{H}_1)$ in legitimate and under attack, respectively are known a priori, the optimal solution is the CUSUM test [7]. More in detail, in the CUSUM test, at time t_n, we

1. compute the LRT
$$\ell_n = \log \frac{P(\boldsymbol{x}_n|\mathcal{H}_0)}{P(\boldsymbol{x}_n|\mathcal{H}_1)} \,, \tag{2}$$
2. update the test variable $v_n = \max\{v_{n-1} + \ell_n, 0\}$, with $v_0 = 0$,
3. check $v_n > \lambda$: if true, an alarm is raised and we consider as detected change time, $t^\star = t_n$.

Indeed, assuming the knowledge of both $P(\boldsymbol{x}_n|\mathcal{H}_0)$ and $P(\boldsymbol{x}_n|\mathcal{H}_1)$ may be unrealistic in many practical applications. First, it is well-known that in general conditions it is not trivial to have a robust model of the UWAC from which to extract the PDF needed for the LRT.

We propose then a data-driven based on NNs. In particular, we consider both a deep NN solution and a RNN approach. As discussed in [15], a NN with a sufficiently complex architecture and a sufficiently large dataset achieves the same performance of the LRT, in terms of false alarm and missed detection probability. On the other hand, it is not feasible to consider all the - possibly infinite - past inputs to the network. Thus at each step, we feed to the network with a finite window of observations, $\boldsymbol{W}_n = [\boldsymbol{x}_{n-L}, \ldots, \boldsymbol{x}_n]$.

Before introducing RNNs we briefly review deep feedforward NNs, focusing on classification tasks. For this case, the aim of a NN is to implement a test function $f(\cdot)$, such that $f(\boldsymbol{x}_n) = 0$ when $\boldsymbol{x}_n \in \mathcal{H}_0$ and $f(\boldsymbol{x}_n) = 1$ when $\boldsymbol{x}_n \in \mathcal{H}_1$. Deep NNs are composed of a series of layers, each containing several neuron units. A NN is said to be fully connected and feedforward when the input of a neuron is the collection of the output of all the neurons from the previous layers.

The output of the k-th neuron of the q-th layer can be computed as

$$y_k^{(q+1)} = \sigma^{(q)}\left(\boldsymbol{w}_k^{(q)} \boldsymbol{y}^{(q)} + b_k^{(q)}\right) \tag{3}$$

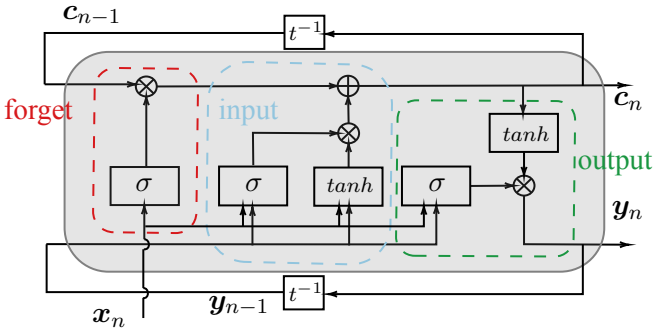


Figure 2: Model of a LSTM with forget (red), input (blue), and output gate (green).

where $\sigma^{(q)}(\cdot)$ 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.

Finally, considering the (single-node) output of the last layer $y_1^{(Q)}$, we choose function f as $f(\boldsymbol{x}) = 1$ if $y_1^{(Q)} \geq \lambda$ and $f(\boldsymbol{x}) = 1$ otherwise, where λ is set priori depending on a target false alarm probability value, P_{FA}. Notice that by increasing λ we reduce P_{FA} and increase the missed detection probability P_{MD}. The NN is optimized (i.e., trained) using algorithms such as adaptive moment estimation (ADAM), by setting as target $y^\star = 0$ if the input was from class $\mathcal{H}_0$ and $y^\star = 1$ if, instead, the input was drawn from $\mathcal{H}_1$.

In RNNs, each output is a function of the previous ones. We focus on long short-term memory RNN (LSTM). With respect to the original RNNs, they have a set of cell states, where $\boldsymbol{c}_n$ is the state at time t_n and their architecture is composed of three gates, the *forget*, the *input*, and the *output* gate. Indeed, by exploiting the cell state, it is possible for RNN to learn the correlation between subsequent samples. In these terms it is possible to compare the cell state to the v_n test variable of the CUSUM. A sketch of a LSTM neuron at the n-th stage (i.e., after the processing of $n-1$ inputs), is reported in Fig. 2.

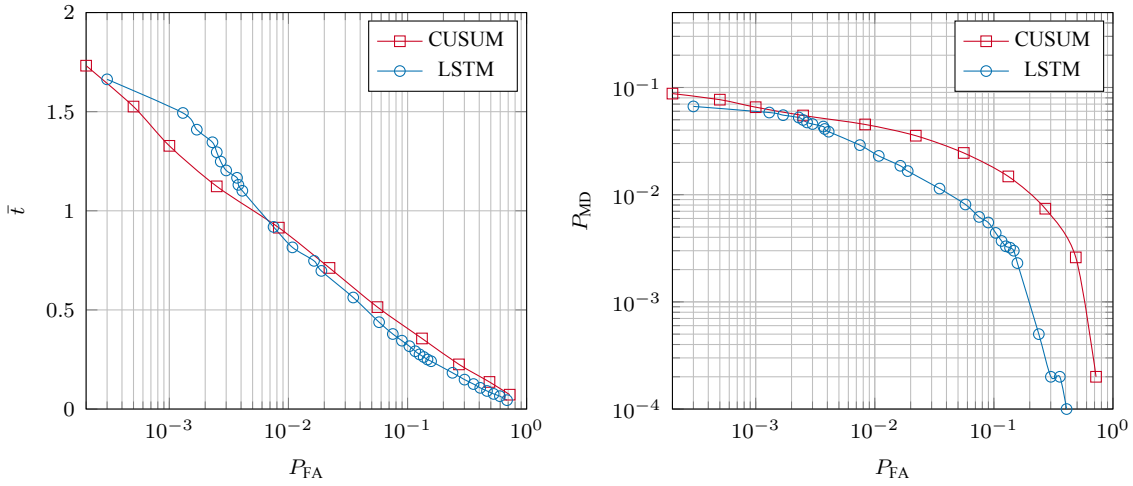
The role of the forget gate is to regulate how much the past cell state will influence the next. In other terms, the LSTM decides which information has to be forgotten and which has to be kept in memory. The input layer instead updates the current state, taking as input both the actual input, $\boldsymbol{x}_n$, and the past output, $\boldsymbol{y}_{n-1}$ that behaves like a hidden state. Finally, the output layer compares the updated state $\boldsymbol{c}_n$ with $\boldsymbol{x}_n$ and $\boldsymbol{y}_{n-1}$ to compute the current output. Blocks $\sigma(\cdot)$ indicate sigmoid activation functions, while $\tanh(\cdot)$ is the hyperbolic tangent. The training procedure for RNNs is identical to the one for deep NNs, but after each training sequence ends all the cell states are reset.Finally, we add a dense single-neuron layer, that output a value, that is thresholded to obtain the final decision. More details about both NNs and RNNs can be found in [16].

4. NUMERICAL RESULTS

We evaluated the performance of the proposed solution by considering both a dataset with Gaussian features and an underwater channel, processed by using data collected during a sea experiment.

Concerning the Gaussian dataset, we consider sequences of $N = 30$ samples where if $t < \tilde{t}$, $x_n \sim \mathcal{N}(0,1)$ while after the attack, i.e., $t \geq \tilde{t}$, $x_n \sim \mathcal{N}(3,1)$. The change time is drawn uniformly within the sequence as $\tilde{t} \sim \mathcal{U}([11,20])$. We generated a total of $40\ 000$ sequences, using the 60% for training, 15% for training, and 25% for testing.

We designed the RNN to have 2 layers, a LSTM layer with 16 neurons, and a single-neuron



(a) Average decision delay *(b) Missed detection*

Figure 3: Performance achieved for the Gaussian dataset by the CUSUM (red squares) and the LSTM (blue circles).

dense layer with a sigmoid activation function. We used as a loss function the binary cross-entropy. At each step, we feed to the RNN the measurements' window $\boldsymbol{W}_n = [x_{n-L}, \ldots, x_n]$ with $L = 10$ samples. To cope with the variable input size, we considered a proper zero-padding and included to the RNN a masking layer.

As a performance metric, we consider the average decision delay

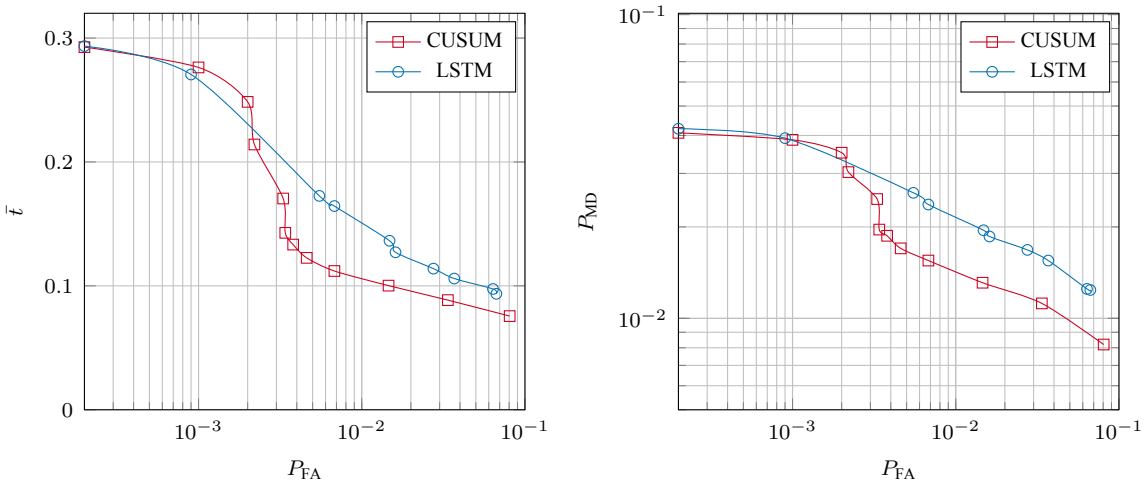
$$\bar{t} \triangleq \frac{1}{N} \sum_i \left(\tilde{t}_i - t_i^\star \right) \ , \tag{4}$$

where $\tilde{t}_i$ and $t_i^\star$ are respectively measured and the actual change time for the i-th testing sequence. We remark that, for infinitely long sequences, the tests will always raise an alarm. In our case, not all the tests raised an alarm before the sequence ended. Thus, we have to consider also the missed detection probability, computed as the average number of sequences where no alarm is raised (even if an attack was actually present). The results are compared with the CUSUM test, detailed in Section 3.

Fig. 3 shows the obtained results, as a function of the user-defined false alarm. Indeed both the decision delay and the missed detection probability decrease as the P_{FA} increases: thus by setting a low threshold, the test becomes more sensible and promptly reacts to both the attack, achieving a lower $\bar{t}$ and P_{MD}, but also to outliers on the legitimate distribution, thus also getting a high P_{MD}. The performance achieved by the RNN are comparable to the optimal ones obtained by the CUSUM test. We remark that, when using the CUSUM test, we assume that the defender has the full distribution of both the legitimate and the under-attack, rather than a set of samples.

The experimental dataset has been collected from an experiment performed in January 2022 in Eilat, Israel. To increase the dataset size, we performed a technique, where, first we estimate the cumulative distribution function (CDF) of each feature from the experimental dataset, by using kernel density estimation with Gaussian kernels. Next, we generate a new dataset where each feature has the same distribution of (estimated) experimental counterpart, by using inverse sampling. More details about both the experiment and the augmented dataset generation can be found in [4].

We generate the input sequence analogously to the Gaussian case, randomly generating $\tilde{t} \sim \mathcal{U}([11, 20])$, but choosing for $\mathcal{H}_0$ and $\mathcal{H}_1$ channel measurements collected from signals transmitted by different receivers. Notice that in this case input sample $\boldsymbol{x}_n$ is now a vector,



(a) Average decision delay *(b) Missed detection*

Figure 4: Performance achieved for the underwater dataset by the CUSUM (red squares) and the LSTM (blue circles).

containing 4 feature measurements. [1] We generated a total of $40\,000$ sequences, using the 60% for training, 15% for training, and 25% for testing. The LSTM network architecture is the same as the one used for the Gaussian data.

Fig. 4 shows the results obtained from the underwater dataset. Both $\bar{t}$ and P_{MD} exhibit a trend similar to that of the Gaussian dataset. Even in this more complex scenario, the CUSUM and the LSTM-based tests achieve almost the same performance, with the CUSUM performing slightly better at high P_{FA} values.

5. CONCLUSIONS

In the context of PLA for UWACs, we tackled the problem of detecting the start of an attack, with the least possible delay. After having properly introduced the problem, and noticing that, typically after the attack start, Eve keeps transmitting fake signals, we framed it as a quickest detection problem. While the CUSUM has been proven to be optimal for the considered scenario, it still requires the knowledge of both the legitimate and under-attack feature distribution, which, in practice, is not known in advance. Hence, in this paper, we resort to LSTMs. These RNNs are equipped with memory, and, differently from deep NNs, they take advantage of the stationarity in our problem.

We tested the performance of the proposed strategies using both a scalar Gaussian dataset, and data collected from a sea experiment. In the latter case, at each considered instant, a vector of 4 features, i.e., number of taps, average tap power, RMS delay, and smoothed received power. Results show in both cases it is possible to achieve the same performance of the CUSUM test, obtained assuming the knowledge of the prior distribution.

6. ACKNOWLEDGMENTS

This work was sponsored in part by the NATO Science for Peace and Security Programme under grant no. G5884 (SAFE-UComm).

[1] Before feeding the input sequence to the LSTM, we flatten the vector.

REFERENCES

[1] M. Bloch and J. Barros, *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[2] R. Diamant, P. Casari, and S. Tomasin, "Cooperative authentication in underwater acoustic sensor networks," *IEEE Trans. on Wirel. Commun.*, vol. 18, no. 2, pp. 954–968, Dec 2019.

[3] L. Bragagnolo, F. Ardizzon, N. Laurenti, P. Casari, R. Diamant, and S. Tomasin, "Authentication of underwater acoustic transmissions via machine learning techniques," in *Proc. of COMCAS*, 2021, pp. 255–260.

[4] F. Ardizzon, R. Diamant, P. Casari, and S. Tomasin, "Machine learning-based distributed authentication of UWAN nodes with limited shared information," in *Proc. of UComms*, 2022, pp. 1–5.

[5] P. Casari, F. Ardizzon, and S. Tomasin, "Physical layer authentication in underwater acoustic networks with mobile devices," in *Proc. of WUWNet*, 2022, pp. 1 – 8.

[6] E. S. Page, "Continous inspection schemes," *Biometrika*, vol. 41, no. 1-2, pp. 100–115, June 1954.

[7] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *Ann. Statist.*, vol. 14, no. 4, pp. 1379 – 1387, Dec. 1986.

[8] S. Li, Y. Yılmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Trans. on Smart Grid*, vol. 6, no. 6, pp. 2725–2735, Dec. 2015.

[9] M. N. Kurt, Y. Yılmaz, and X. Wang, "Distributed quickest detection of cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 2015–2030, Feb. 2018.

[10] S. Tomasin, "Consensus-based detection of malicious nodes in cooperative wireless networks," *IEEE Commun. Lett.*, vol. 15, no. 4, pp. 404–406, Mar. 2011.

[11] J. Dehning, J. Zierenberg, F. P. Spitzner, M. Wibral, J. P. Neto, M. Wilczek, and V. Priesemann, "Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions," *Science*, vol. 369, no. 6500, May 2020.

[12] L. Xie, S. Zou, Y. Xie, and V. V. Veeravalli, "Sequential (quickest) change detection: Classical results and new directions," *IEEE J. Sel. Areas in Info. Theory*, vol. 2, no. 2, pp. 494–514, June 2021.

[13] K. Pelekanakis, S. A. Yıldırım, G. Sklivanitis, R. Petroccia, J. Alves, and D. Pados, "Physical layer security against an informed eavesdropper in underwater acoustic channels: Feature extraction and quantization," in *Proc. of UComms*, 2021.

[14] G. Lorden, "Procedures for reacting to a change in distribution," *Ann. Math. Stat.*, vol. 42, no. 6, pp. 1897 – 1908, Dec. 1971.

[15] A. Brighente, F. Formaggio, G. M. Di Nunzio, and S. Tomasin, "Machine learning for in-region location verification in wireless networks," *IEEE J. Sel. Areas in Comm.*, vol. 37, no. 11, pp. 2490–2502, 2019.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.