Explainability of Neural Networks for Symbol Detection in Molecular Communication Channels

Jorge Torres Gómez, *Senior Member, IEEE*, Pit Hofmann, *Graduate Student Member, IEEE*, Frank H.P. Fitzek, *Senior Member, IEEE*, and Falko Dressler, *Fellow, IEEE*

***Abstract*—Recent molecular communication (MC) research suggests machine learning (ML) models for symbol detection, avoiding the unfeasibility of end-to-end channel models. However, ML models are applied as black boxes, lacking proof of correctness of the underlying neural networks (NNs) to detect incoming symbols. This paper studies approaches to the explainability of NNs for symbol detection in MC channels. Based on MC channel models and real testbed measurements, we generate synthesized data and train a NN model to detect of binary transmissions in MC channels. Using the local interpretable model-agnostic explanation (LIME) method and the individual conditional expectation (ICE), the findings in this paper demonstrate the analogy between the trained NN and the standard peak and slope detectors.**

***Index Terms*—Explainable AI, individual conditional expectation, local interpretable model-agnostic explanation, machine learning, molecular communication, neural network, testbed**

I. Introduction

RESEARCH in molecular communication (MC) targets new symbol detection methods avoiding the use of end-to-end channel models and the estimation of their parameters. The lack of closed-form expressions for MC channels makes it impractical to design detectors, as their optimal functioning depends on the end-to-end channel impulse response (CIR). Even with closed-form expressions for the CIR, still, variable parameters, like the distance between emitters and receivers, prevent setting detection thresholds to distinguish high from low levels in, e.g., on-off keying (OOK) transmissions [1], [2]. Circumventing these impediments, the research community is actively studying the use of machine learning (ML) methods as an expedient way to build near-optimal detectors [3].

In the literature, supervised learning using neural networks (NNs) is becoming the de-facto standard for implementing detectors. Architectures like artificial neural networks (ANN), recurrent neural networks (RNN), convolutional neural networks (CNN), or bidirectional RNN are trained offline with a large number of received symbols. The models are then deployed for online inference, here, symbol detection. Such neural network (NN)-based solutions are applied to the detection of pH levels controlled by Escherichia Coli bacteria [4], using pumping machines in fluids [5], [6], and in free-diffusion environments [7]–[9]. In addition, the k-means clustering method is reported, evaluating the detector threshold in free-diffusion environments [10].

Although the effectiveness of applying NN architectures for symbol detection is evident, a proof of effectiveness, i.e., their explainability, is missing. As Huang et al. [3] mentioned, the explainability in MC systems is even more vital than in conventional communication systems. ML models are applied as black boxes without taking care of proof of correctness for the long-term functionality. The matter becomes relevant in developing provable models [11], where trustworthiness is crucial to deploy in-vivo systems for healthcare applications like targeted drug delivery, or cancer cell detection [3].

In this paper, we study the explainability of NN-based architectures for symbol detection in MC channels. As for the NN explainability, we apply the local interpretable model-agnostic explanation (LIME) method and the individual conditional expectation (ICE) [12], also for ease of illustration. For validation, we target a real testbed scenario as well as an end-to-end channel model of a point transmitter, a free diffusion channel, and a spherical absorbing receiver. Findings illustrate the analogy between the trained NN and standard detectors as the peak or the slope detectors, depending on extracted features from the received signal. Based on such analogies, this paper aims to introduce explainable methods searching for mathematical models to later provide assurance of correctness on the use of NNs for symbol detection.

Our key contributions can be summarized as follows:

- We study the use of ML models for symbol detection in MC channels, also training on real testbed data;
- we, for the first time, study the explainability of NN models used for MC; and
- we show results demonstrating the analogy between the trained NN and standard peak and slope detectors.

II. System Model

A. Synthesizing Data

We synthesize two databases to train the NN model using the end-to-end channel models in [13] and the testbed presented in our previous work accounting for a more realistic scenario (see

Jorge Torres Gómez and Falko Dressler are with the School for Electrical Engineering and Computer Science, TU Berlin, Berlin, Germany, email: {torres-gomez, dressler}@ccs-labs.org.

Pit Hofmann and Frank H.P. Fitzek are with the Deutsche Telekom Chair of Communication Networks, Technische Universität Dresden, Dresden, Germany; F. Fitzek is also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI), Dresden, Germany, email: {pit.hofmann, frank.fitzek}@tu-dresden.de.

This work was supported in part by the project MAMOKO funded by the German Federal Ministry of Education and Research (BMBF) under grant numbers 16KIS0917 and by the project NaBoCom funded by the German Research Foundation (DFG) under grant number DR 639/21-2. This work was also supported by the German Research Foundation (DFG) as part of Germany's Excellence Strategy—EXC 2050/1—Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden under project ID 390696704 and the Federal Ministry of Education and Research (BMBF) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-life, grant number 16KISK001K.

TABLE I
RANGE OF PARAMETERS USED WITH THE SYNTHETIC CIR AS IN EQ. (2).

Parameter	Variable	Value
Molecules per emission	N_{Tx}	10^4
Distance	d	200 nm
Receiver radius	r_{Rx}	50 nm
Diffusion coefficient	D	$10^{-10}\,\mathrm{m}^2/\mathrm{s}$
Bit duration	T_b	1 ms (low-interference regime)
		100 ns (high-interference regime)
Sampling time	T_s	100 ns (low-interference regime)
		1 ns (high-interference regime)

Fig. 1) [14]. For the end-to-end channel model, we employ the standard case point transmitter-free diffusion channel-transparent spherical receiver, where the total number of received molecules follows a Poisson distribution as [13, Eq. (74)]

$$N_{\mathrm{Rx}} \sim \mathcal{P}\left(N_{\mathrm{Tx}} h(t)\right), \tag{1}$$

where $h(t)$ is the CIR as [13, Eq. (34)]

$$h(t) = \frac{V_{\mathrm{Rx}}}{(4\pi D t)^{\frac{3}{2}}} e^{-\frac{d^2}{4Dt}}, \tag{2}$$

where N_{Tx} and N_{Rx} denote the total number of transmitted and received molecules, respectively, V_{Rx} is the receiver's volume, D is the diffusion coefficient of the molecules, d is the distance between the transmitter and the receiver, and t is time. We remark that the Poisson distribution in Eq. (2) results sufficiently accurate whenever the amplitude of the CIR is less than 10^{-1}, which is valid in our case. We also remark that the Eq. (2) is valid whenever the total number of received samples is independently distributed over time [13]. The independent assumption is also reasonable in our case as we avoid any non-linear behavior at the receiver side (e.g., we avoid the receiver occupancy effect), and the channel is linear and time-invariant. Using the CIR in Eq. (2) with the parameters listed in Table I, we generate synthetic data for 100 emissions (the emission of molecules accounts for a one, while no emissions account for a zero). Parameters provided in Table I follow values in [13].

We also synthesize data using the testbed illustrated in Fig. 1 (see [14] for further details). The transmitter is equipped with a sprayer, and the receiver with a sensor located at a distance of 1 m. The sprayer releases a total number of $3.92{\times}10^{21}$ ethanol molecules into the air with a diffusion coefficient $D = 0.84 \times 10^9\,\mathrm{m}^2/\mathrm{s}$. The dissolution has a mass ratio of one part of ethanol (H_3CCH_2OH) into four parts of water as $m_{H_3CCH_2OH}/m_{H_2O} = 1/4$. Molecule transmissions are supported with a standard fan, producing a channel with drift of the average speed $v = 3.5\,\mathrm{m/s}$.[1] All used devices and sensors are commercial-off-the-shelf ones – reprogrammable, reproducible, and adaptable for different research approaches.

We use the testbed to generate data with recorded pulse transmissions at the chemical sensor in Fig. 1. We averaged over 40 emissions of 1's, where each emission lasts for

[1]The airflow speed was measured using an anemometer Airflow Instruments LCA301.

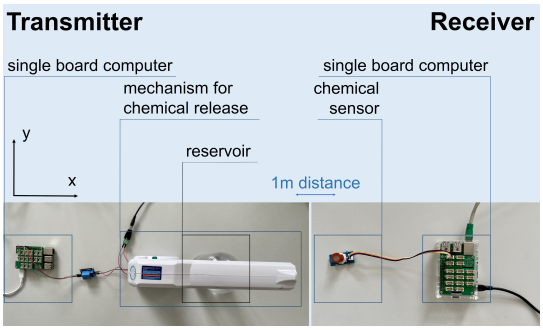


Fig. 1. Molecular SISO communication system testbed [14].

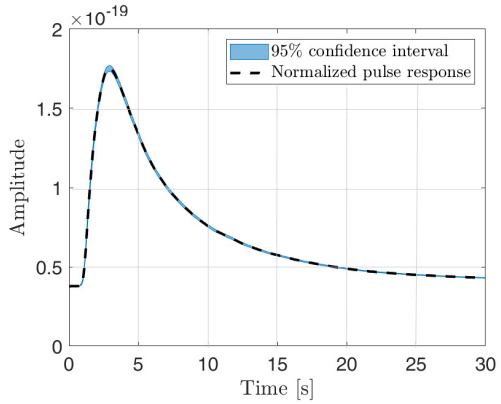


Fig. 2. Average and 95 % confidence interval (blue area) for the normalized received pulses after averaging 40 emissions in the testbed.

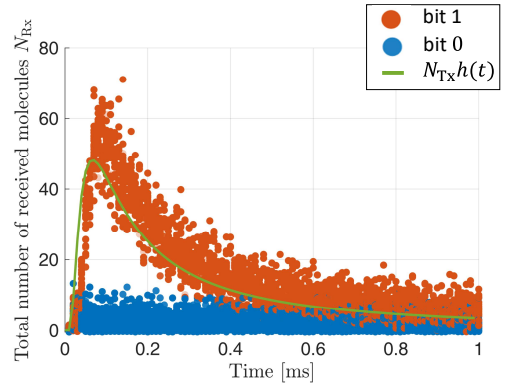
2 s and samples are collected at the sensor node along 30 s with sampling time 0.1 s.[2] Fig. 2 illustrates the average pulse response after normalizing it with the total of emitted molecules. The plot also shows the 95 % confidence interval (blue area). This plot evidences that the average pulse sequence accurately describes transmissions over the channel as we observe a low variability of samples. Using the estimated pulse response, we integrate the testbed to produce synthetic data using it as the argument of the Poisson distribution in Eq. (1) and randomly produce an OOK sequence of arbitrary length. We use the Poisson distribution aiming to include some randomness in the amplitude of received pulses. We remark that we perform sufficiently large time intervals between the bit-1's (as 30 s) to avoid saturation at the receiver sensor, as produced by a high concentration of molecules with consecutive pulse emissions.

B. Neural Networks Architecture

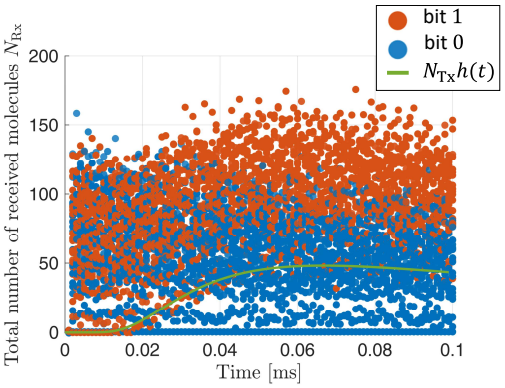
To predict the transmitted bits, we implement a fully connected feed-forward NN in Matlab with two layers for classification, where the first layer has 10 outputs and the second layer produces the binary output. The first layer uses the rectified linear unit (ReLU) and the second layer the Softmax activation functions.

The NN is trained to minimize the cross-entropy loss function to find the NN parameters [15]. Data for training are synthetically generated in Matlab using the above Eq. (1)

[2]The dataset for the testbed measurements and the Matlab processing code are available at IEEE DataPort at http://ieee-dataport.org/11110.

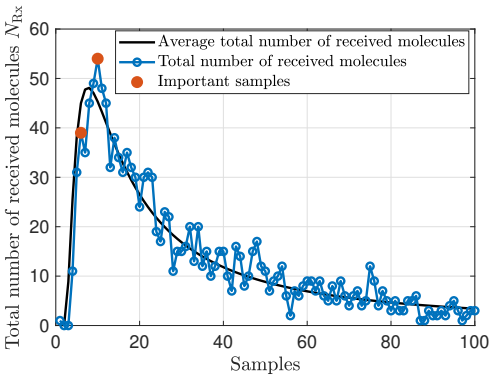


(a) Low-interference regime

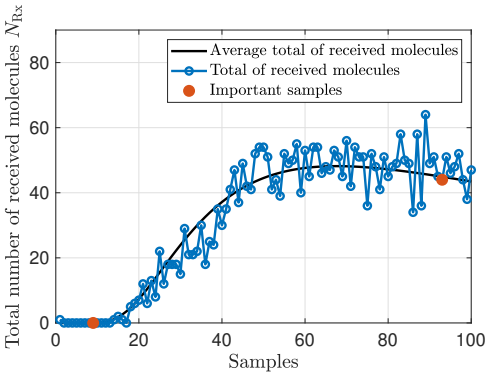


(b) High-interference regime

Fig. 3. MOL-eye diagram.



(a) Low-interference regime



(b) High-interference regime

Fig. 4. Important samples used by the NN detector according to LIME for low- and high-interference regime. We recall that some samples at the beginning of reception are zero due to the low probability of molecules' arrival.

with the argument given as in Eq. (2) on the one hand, and with the average received molecules measured with the testbed on the other hand. The NN is trained with $50\,\%$ and evaluated with the remaining $50\,\%$ of the generated data.

III. Explaining the Neural Networks Behavior

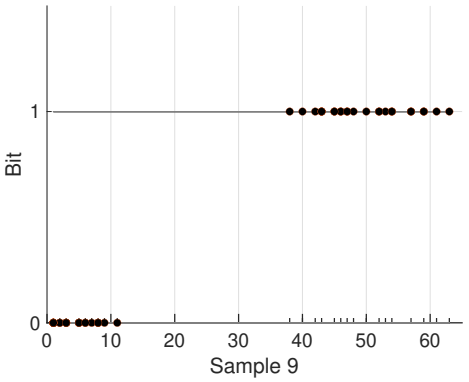
In this Section, we provide explanations based on the synthetically generated data using Equations (1) and (2) on the one hand, and with the testbed average pulses in Fig. 2 on the other hand. As for the synthetic data, we train the NN under two regimes, low- and high-interference levels. As depicted in the MOL-eye diagram [16] in Fig. 3a, the low-interference diagram exhibits a wide open eye in contrast to the high-interference one in Fig. 3b. The level of interference is tuned with the bit duration T_b – the low level is the case when $T_b = 1\,\mathrm{ms}$, where the amplitude on the tail is less than $1\,\%$ the amplitude on the peak. The high level of interference is produced when $T_b = 100\,\mathrm{ns}$, where the peak amplitude, at $T_p = \frac{d^2}{6D} \approx 66\,\mathrm{ns}$ [13], is located close to the bit duration and the total number of received molecules from the previous emission are the $94\,\%$ of the actual transmission peak amplitude.

To explain the trained NN model, we apply the local interpretable model-agnostic explanation (LIME) method with the Matlab function `lime()`. The LIME method fits a linear model while identifying the most valuable predictors (inputs samples) to distinguish the ones from the zeros in the two regimes. The `lime` function implements the group orthogonal matching pursuit (OMP) algorithm for the predictors selection [17]. It estimates the most valuable predictors solving an equivalent optimization problem with a least square regression formulation [18].

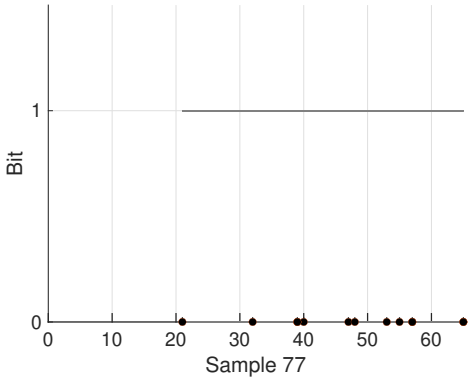
Results of the LIME method are illustrated in Fig. 4, where we used the same closed form expression $N_{\mathrm{Tx}}h(t)$ as in Equations (1) and (2) for the fairness of comparison. The trained NN model utilizes those samples located at the peak of the received symbols in the low-interference level regime (see Fig. 4a). As expected, those samples are more distant from zero, thus producing a better distinction between the two emissions. Meanwhile, in the high-interference regime (see Fig. 4b), the NN method employs samples located at the beginning of symbols and at the peak to distinguish emissions.

We depict in Fig. 5 the ICE plot for the two interference regimes. The ICE plot illustrates the correspondence between the predictor variable (input molecules) and the predicted class (zero or one) for a given observation (Sample 9 in Fig. 5a), for instance) [12]. In the horizontal axis are the amplitude values (some of them are superposed) taken among the various symbols, and in the vertical axes, it is the produced output as the bit `1` or bit `0`. This plot visualizes a relation between the sample value and the NN decision. As Fig. 5a depicts, in the low-interference regime the NN outputs a `1` whenever the sample is higher than 39 molecules, and `0` whenever it is less than 11 molecules. Based on the two plots Figures 4a and 5a, we can readily interpret that the NN is operating as a peak-detector for the low-interference regime, where samples in the peaks are compared to a threshold, in this way distinguishing emissions in one from zero (see [19, Sec. V.A.2.d]).

Interpreting the NN operation as a peak-detector, then we

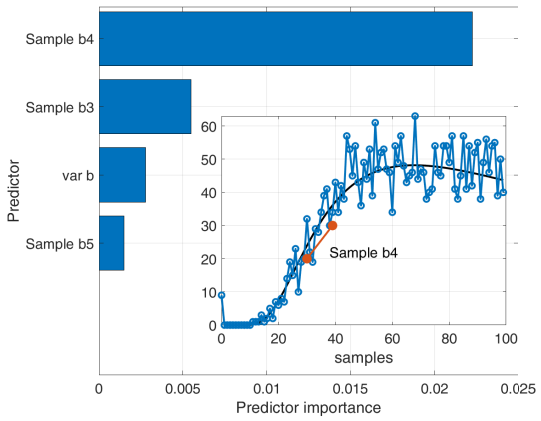


(a) Low-interference regime

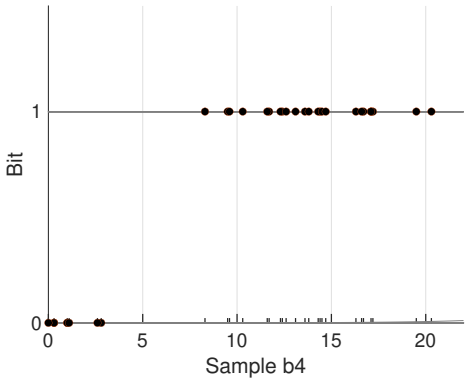


(b) High-interference regime

Fig. 5. ICE plot of the NN detector for low- and high-interference regimes.



(a) LIME method



(b) ICE plot

Fig. 6. Interpretability for the slope detector in [5]. Most important samples are distinguished with LIME method and the corresponding ICE plot.

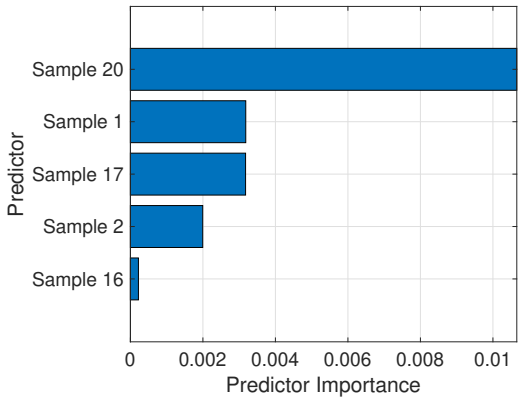
can use the corresponding equations to evaluate the bit error rate (BER) (see [20, Sec. III]) with the threshold given by the Fig. 5a (around 30 molecules). However, in the low-interference regime the BER can be readily evaluated to be less than 10^{-5}, see [20, Fig. 15], as we are considering any interference noise and negligible inter-symbol interference (ISI).

In the high-interference regime, it results non-meaningful to interpret the ICE plot in Fig. 5b as all the samples score zero regardless of their value. In this case, it is needed to apply a different mechanism to interpret the rule implemented by the NN. In our opinion, the NN might be implementing a slope detector, as it uses samples at the beginning and at the peak of incoming symbols to perform detection (cf. Fig. 4b).

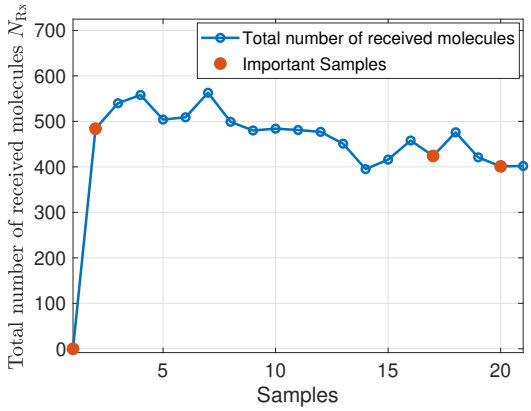
In the high-interference regime, we also implement the slope detector due to its effectiveness to reduce the ISI [5], [21]. We follow the detector conceived in [5], where the NN is trained with features as the slope between segments, 10 samples each, of the received signals and the variance of samples as well (see [5, Appendix]). Fig. 6a illustrates the most important predictors used by the NN model, where samples `b's` denote the slope of the received sequence, and `var b` denotes the variance of these samples. As a result of this plot, the most important sample is `Sample b4`, which represents the steepest slope from segments in the transition time interval between the low and the high-level, i.e., located between 20 samples and 40 samples in Fig. 4. Besides, the ICE plot in Fig. 6b exhibits a well-distinction to score the detection of the ones; e.g., samples with a slope higher than six units are identified as ones. Using these features, this NN will behave mostly as a slope detector when comparing the steepest slope of the incoming symbol to a threshold. Coincidentally, the slope detector has been reported as a solution to reduce the impact of ISI in MC channels [22]. The peak value of the derivative is closer to the beginning of the symbol than the peak of the CIR, which allows it to reduce the impact of ISI.

When interpreting the NN as a slope detector, we evaluate its performance with the same closed-form expression as the threshold detector but calculating its input signal-to-noise ratio (SNR) differently [23]. As the slope detector evaluates the difference between two samples at T_b and 0, as follows from Fig. 4b, the signal's power at the output of the slope detector approximates as $(N_{\mathrm{Tx}}h(T_b))^2$. Besides, the noise's power can be readily evaluated as $2N_{\mathrm{Tx}}h(T_b)s[k]$ [23] modeling the received noise like a Gaussian process [13, Eq. (89)]. Then, after evaluating the ratio between the signal and noise powers, the $\mathrm{SNR} = 0.5N_{\mathrm{Tx}}h(T_b) \approx 31\,\mathrm{dB}$ using the parameters in Table I and Eq. (2) to evaluate $h(T_b)$. Finally, according to the plot in [20, Fig. 15] the $\mathrm{BER} \approx 4 \times 10^{-2}$.

Finally, we plot the results for the LIME method in Fig. 7 using the testbed-generated sequence with pulses. We aim to illustrate results for a more realistic scenario with experimental data. In this case, we construct a high-interference regime, where the symbol time is $T_b = 2\,\mathrm{s}$, with the corresponding superposition of pulse transmissions. The LIME plot exhibits that the most important samples to perform detection are at the beginning and at the end of the symbol, which is similar to our



(a) LIME method



(b) Important samples

Fig. 7. Interpretability for the NN detector using testbed data.

simulation results in Fig. 4b. Although we do not plot here, the ICE plot looks also similar to the case in Fig. 5b, which does not allow interpreting a rule. Similarly to the simulator results for the high-interference regime, we also suggest that the NN is implementing a slope detector to distinguish the ones from the zeros in the testbed real case scenario.

IV. Conclusion and Future Research Directions

Research reports symbol detection in MC channels using a NN due to its flexibility despite unknown channel parameters. With the help of a testbed, we trained a NN for symbol detection in MC. We then investigated the explainability of NN models, opening ways to provide proof of accuracy as detectors. Findings illustrate that our NN operates as intuitively expected to detect emissions in ones from zero. In the low-interference regime, it behaves as a peak detector. Besides, in the high-interference regime and using the slope of input samples as features, the NN mostly uses the slope located at the transitions, thus, performing as a slope detector. Our findings also illustrate that the NN emulates a low-complex mechanism to decode the received signal, as only a few samples are selected to distinguish emissions. Future work will be conducted to analyze the optimality of the NN-based detectors with the threshold.

References

[1] M. S. Kuran, H. B. Yilmaz, I. Demirkol, N. Farsad, and A. Goldsmith, "A Survey on Modulation Techniques in Molecular Communication via Diffusion," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 7–28, Jan. 2021.

[2] S. Bhattacharjee, M. Damrath, L. Stratmann, P. A. Hoeher, and F. Dressler, "Digital Communication Techniques in Macroscopic Air-Based Molecular Communication," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 8, no. 4, pp. 276–291, Dec. 2022.

[3] Y. Huang, F. Ji, Z. Wei, M. Wen, and W. Guo, "Signal Detection for Molecular Communication: Model-Based vs. Data-Driven Methods," *IEEE Communications Magazine*, vol. 59, no. 5, pp. 47–53, May 2021.

[4] F. Vakilipoor, D. Scazzoli, F. Ratti, G. Scalia, and M. Magarini, "Hybrid deep learning-based feature-augmented detection for molecular communication systems," in *ACM NANOCOM 2022*, Barcelona, Spain: ACM, Oct. 2022.

[5] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.

[6] L. Sun and Y. Wang, "CTBRNN: A Novel Deep-Learning Based Signal Sequence Detector for Communications Systems," *IEEE Signal Processing Letters*, vol. 27, pp. 21–25, 2020.

[7] S. Sharma, D. Dixit, and K. Deka, "Deep Learning based Symbol Detection for Molecular Communications," in *IEEE ANTS 2020*, New Delhi, India: IEEE, Dec. 2020.

[8] G. H. Alshammri, W. K. M. Ahmed, and V. B. Lawrence, "Receiver Techniques for Diffusion-Based Molecular Nano Communications Using an Adaptive Neuro-Fuzzy-Based Multivariate Polynomial Approximation," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 3, no. 4, pp. 140–159, Sep. 2018.

[9] X. Qian and M. Di Renzo, "Receiver Design in Molecular Communications: An Approach Based on Artificial Neural Networks," in *IEEE ISWCS 2018*, Lisbon, Portugal: IEEE, Aug. 2018.

[10] X. Qian, M. Di Renzo, and A. Eckford, "K-Means Clustering-Aided Non-Coherent Detection for Molecular Communications," *IEEE Transactions on Cognitive Communications and Networking*, vol. 69, no. 8, pp. 5456–5470, Aug. 2021.

[11] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Toward verified artificial intelligence," *Communications of the ACM*, vol. 65, no. 7, pp. 46–55, Jul. 2022.

[12] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[13] V. Jamali, A. Ahmadzadeh, W. Wicke, A. Noel, and R. Schober, "Channel Modeling for Diffusive Molecular Communication - A Tutorial Review," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1256–1301, Jul. 2019.

[14] P. Hofmann, J. Torres Gómez, F. Dressler, and F. H. P. Fitzek, "Testbed-based Receiver Optimization for SISO Molecular Communication Channels," in *IEEE BalkanCom 2022*, Sarajevo, Bosnia and Herzegovina: IEEE, Aug. 2022, pp. 120–125.

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

[16] M. Turan, M. S. Kuran, H. B. Yilmaz, C.-B. Chae, and T. Tugcu, "MOL-eye: A new metric for the performance evaluation of a molecular signal," in *IEEE WCNC 2018*, Barcelona, Spain: IEEE, Apr. 2018.

[17] A. Lozano, G. Swirszcz, and N. Abe, "Group Orthogonal Matching Pursuit for Logistic Regression," in *AISTATS 2011*, Fort Lauderdale, FL, Nov. 2011, pp. 452–460.

[18] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[19] M. Kuscu, E. Dinc, B. A. Bilgin, H. Ramezani, and O. B. Akan, "Transmitter and Receiver Architectures for Molecular Communications: A Survey on Physical Design With Modulation, Coding, and Detection Techniques," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1302–1341, Jul. 2019.

[20] X. Qian, M. D. Renzo, and A. W. Eckford, "Molecular Communications: Model-Based and Data-Driven Receiver Design and Optimization," *IEEE Access*, vol. 7, pp. 53 555–53 565, Jan. 2019.

[21] Y. Huang, X. Chen, M. Wen, L.-L. Yang, C.-B. Chae, and F. Ji, "A Rising Edge-Based Detection Algorithm for MIMO Molecular Communication," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 523–527, Apr. 2020.

[22] Y. Hao, G. Chang, Z. Ma, and L. Lin, "Derivative-Based Signal Detection for High Data Rate Molecular Communication System," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1782–1785, Sep. 2018.

[23] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*, 4th ed. New York City, NY: McGraw-Hill, 2002, p. 850.