Communicating Smartly in Molecular Communication Environments: Neural Networks in the Internet of Bio-Nano Things

Jorge Torres Gómez, *Senior Member, IEEE*, Pit Hofmann, *Graduate Student Member, IEEE*, Lisa Y. Debus, *Graduate Student Member, IEEE*, Osman Tugay Başaran, *Graduate Student Member, IEEE*, Sebastian Lotter, Roya Khanzadeh, *Member, IEEE*, Stefan Angerbauer, Bige Deniz Unluturk, *Member, IEEE*, Sergi Abadal, Werner Haselmayr, Frank H.P. Fitzek, *Fellow, IEEE*, Robert Schober, *Fellow, IEEE*, and Falko Dressler, *Fellow, IEEE*

***Abstract*—Recent developments in the Internet of Bio-Nano Things (IoBNT) are laying the foundation for innovative healthcare applications that envision a network of remotely coordinated nanodevices within the human body to monitor and actuate over potential diseases. However, interconnecting such nanodevices requires communication strategies that can cope with molecular communication (MC) channels, whose complex, stochastic, and dynamic behavior often makes accurate physical modeling infeasible. To explore the limits of nanodevice interconnectivity under these conditions, this survey focuses on data-driven communication strategies for MC systems, with particular emphasis on machine learning (ML) methods and neural network (NN) architectures for a robust and adaptive communication scheme at the nanoscale. Research on NN-enabled MC spans several aspects covered in this survey, including NNs for communication in IoBNT networks, the feasibility of biocompatible NN realization, explainable approaches, and the generation of training datasets. We also include open-source code examples to support reproducible research across key MC scenarios. Finally, we identify emerging challenges, including the need for robust NN architectures, biologically integrated NN modules, and scalable training strategies.**

***Index Terms*—Machine learning, neural networks, deep learning, molecular communication, Internet of Bio-Nano Things**

J. Torres Gómez, L. Debus, O. Tugay Başaran, and F. Dressler are with the School of Electrical Engineering and Computer Science, TU Berlin, Germany, Email: {torres-gomez,debus,basaran,dressler}@ccs-labs.org. R. Khanzadeh, S. Angerbauer, and W. Haselmayr are with the Johannes Kepler University Linz, Austria, Email: {roya.khanzadeh,stefan.angerbauer,werner.haselmayr}@jku.at.P. Hofmann and F. H.P. Fitzek are with the Deutsche Telekom Chair of Communication Networks, Dresden University of Technology, Germany, and the Centre for Tactile Internet with Human-in-the-Loop (CeTI), Dresden, Germany, Email: {pit.hofmann,frank.fitzek}@tu-dresden.de. B. D. Unluturk is with Michigan State University, 775 Woodlot Dr, East Lansing, MI, USA, e-mail: unluturk@msu.edu.S. Abadal is with Universitat Politècnica de Catalunya, Spain, Email: abadal@ac.upc.edu.S. Lotter and R. Schober are with the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany, Email: {sebastian.g.lotter,robert.schober}@fau.de. This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) through the project IoBNT, grant numbers 16KIS1986K & 16KIS1994, the project 6G-life, grant number 16KIS2413K, and the project CommUnity, grant number 16KISS012K. This work was also supported by the German Research Foundation (DFG) through the project NaBoCom, grant number DR 639/21-1, SmartSynch grant number TO 1422/4-1, and the Excellence Strategy – EXC 2050/2, Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI). Further support has been given by the "University SAL Labs" initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic-based systems'. Stefan Angerbauer and Werner Haselmayr acknowledge the Linz Institute of Technology (LIT), Johannes Kepler University at Linz, and the State of Upper Austria (Project LIT-2024-13-SEE-115) for the funds.

I. Introduction

Inspired by Schrödinger's reflections about the question "What is life?" [1], the physics community joined biology to describe constituent components on the boundaries between inanimate matter and life. Today, more communities are joining the realm of biology, including electrical engineers and computer scientists, to enable medical, sensing, and communication applications. Frameworks like the Internet of Bio-Nano Things (IoBNT) [2] can only be realized through truly interdisciplinary research leveraging expertise from engineering, computer science, and life sciences. Focusing on recent advancements in artificial intelligence (AI) to realize IoBNT applications, this survey takes the journey one step further. We explore the research progress at the intersection of computer science and biology, specifically the use of neural networks (NNs) to enable molecular communication (MC) links and nanonetworks.

The constituent MC links of the IoBNT framework serve as synthetic tools with promising applications in healthcare and industry, though they pose new challenges for practical deployment. As in the case of wireless links, communication over MC channels is unreliable, albeit for distinct reasons. The combined impacts of random molecular mobility, reactive environments, and system geometry render analytical modeling impractical for most realistic scenarios. As such, model-based MC design becomes infeasible in many practical settings, motivating self-learning mechanisms, such as machine learning (ML), and particularly NN architectures, that can account for unknown parameters and changing environmental conditions.

A. Prelude

Today, the research community contributes to the joint field of ML and IoBNT networks in various directions. As indicated in Fig. 1, a growing number of works focus on developing the physical layer (PHY) of IoBNT, while other contributions focus on the medium access control (MAC) and upper layers. Alongside these, other relevant directions arise: The transparency of these models for sensitive applications, such as the healthcare sector; the representativeness of the datasets used to train these models; and the feasibility of NN deployment at the nanoscale level. These ongoing research activities lay the groundwork for numerous new developments, as we unveil in this survey.

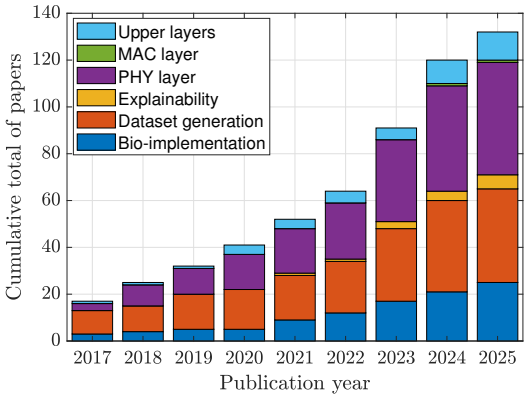


Fig. 1: Trends in contributions related to NN research for the IoBNT.

This research landscape also distinguishes IoBNT from intelligent wireless networks. In wireless systems, ML was introduced after decades of mature communication design; in IoBNT, by contrast, communication architectures are being developed in an era where ML is already a foundational tool. Nevertheless, several lessons transfer: purely modular designs may be suboptimal, and joint optimization of communication blocks can improve performance. Unlike wireless systems, MC links are shaped by diffusion, flow, reactions, and biological variability, requiring models and learning strategies tailored to molecular environments. Together, these advances and challenges frame the central question of this survey: how NNs can support reliable, explainable, and deployable communication in MC-based IoBNT networks.

B. Literature Review Strategy

Tracing the above question within the literature, this survey followed a three-stage search process. First, we searched paper abstracts using the keywords "Neural Network" `AND` ("Molecular Communication" `OR` "IoBNT"). Second, we expanded this set by examining both the references cited by these papers and the works that cited these works. Finally, we curated the resulting collection by selecting papers relevant to biocompatible implementations of NNs and dataset generation. In the first stage, we conducted a comprehensive search across the leading databases IEEE Xplore, ACM Digital Library, ScienceDirect (Elsevier), Springer, and Wiley Online Library. In all cases, our inclusion criteria refer to: (i) The use of NNs as enablers of communication over MC links, which includes the communication design across all layers of the MC channel, i.e., from the PHY layer to the application layer, (ii) biocompatible implementation of NN models, (iii) explainable methods for NNs as applied to MC, and (iv) dataset creation. We listed a total of 397 papers, and adhering to the inclusion criteria outlined above, we filtered these references to 133 published papers, spanning from 2017 to 2025. Our selection aimed to identify representative works that illustrate the evolution of NN applications relevant to the IoBNT framework, ensuring coverage of both foundational and recent contributions across all communication layers.

C. Contributions

Focusing on NN architectures, this survey aims to outline the potential of NNs in IoBNT applications from various perspectives. This paper not only covers the architectures and deployment of NNs but also complementary directions, such as implementation, explainability approaches, and dataset generation for NN model training. The contributions are summarized as follows:

- Deployment of NN architectures for MC links: We discuss in detail the NN architectures reported for the various IoBNT layers, with a primary focus on the PHY layer, as it remains the most mature research area. We aim to provide readers with a holistic view of problems, environments, and NN architectures in each communication layer.
- Code developments for NN-based designs in MC links: We provide illustrative code examples for training and the use of NNs in MC systems. This open-access code encompasses a range of NN architectures across various MC environments, including free-diffusion, flow-based, and vessel-like channels, using both synthetic and testbed-generated datasets. We provide practical insight into the integration of NNs in MC channels.[1]
- Review of potential implementations at the nanoscale level: We summarize state-of-the-art technologies for implementing NN architectures at the nanoscale level. We describe biocompatible technologies such as microfluidic circuits and deoxyribonucleic acid (DNA) chemical reactions, which provide means to run NN modules. We discuss the feasibility of deploying reported NN-solutions in the nanoscale domain.
- Summary of explainable approaches to describe the operation of NNs: We provide a tutorial-style description of explainable methods, offering insights into their application in MC channels. We survey the latest explainable methods studied for MC applications.
- Comprehensive summary of synthetic and testbed-based generation of datasets for training NNs: We elaborate in detail on MC-related datasets used to train NN modules. We also review dataset accessibility, documentation, and usability based on the published code and documentation per dataset.
- New upcoming challenges: We summarize new research directions related to the convergence of NN architectures and MC. We identify open problems in developing, deploying, and training robust NNs, as well as challenges in explaining their operation.

D. Reader's Itinerary

We structured the content to guide the reader for an in-depth examination of the joint topic NN and MC as follows: In Sec. III, we discuss mechanisms by which NNs enable IoBNT networks, which constitutes the central theme of the paper. In Sec. IV, we examine the synthesis of NN modules using biological components. In Sec. V, we review explainability approaches specifically tailored to AI in MC channels. In

[1]We provide the code on two platforms: (i) In the Ocean Code platform, we provide access to the cell-to-cell example developed in Sec. III-A4 under the link https://codeocean.com/capsule/6777864/tree/v1, and (ii) in the GitHub platform we provide access to all code on this paper related to synchronization, decoding, and autoencoding under the link https://github.com/tkn-tub/NN_molecular_communications. Furthermore, we provide the database used to train and test the reported NN modules in [3].

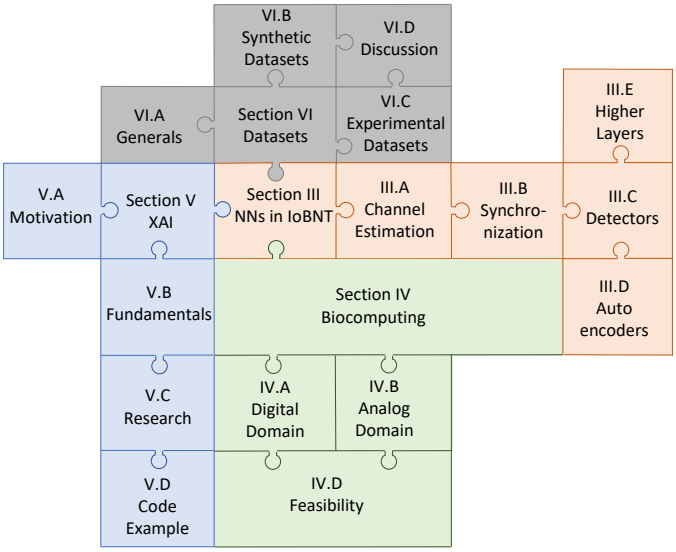


Fig. 2: Mosaic representation of the survey content.

Sec. VI, we present synthetic and testbed-generated datasets for training and evaluating NN modules. Among these sections, the most extensive part is Sec. III, which examines standard communication components (supported by NNs) such as channel estimation, synchronization, detection, and autoencoder designs.

Besides, this survey is organized to accommodate readers with different levels of familiarity with NN and MC. Fig. 2 provides a visual roadmap highlighting the relationships among the main sections and possible independent reading paths.[2] Sec. III forms the core and examines NN architectures for channel estimation, synchronization, detection, and autoencoder design. Readers primarily interested in recent developments may begin with Sections IV and V, which focus on biological implementations and explainability, respectively. Those seeking a gradual introduction can start with Sections III and VI for foundational background before moving on to these more specialized topics. Besides, all sections link back to the core discussion in Sec. III, ensuring continuity across the themes of NNs in MC, biocomputing, explainable AI, and dataset generation.

II. Most Recent Developments in Molecular Communications: A Survey-based Perspective

This section contextualizes recent developments in molecular communication and nanonetworked IoBNT systems within the landscape of related surveys published between 2019 and 2025. Across these surveys, a common trend is the growing need for adaptive communication methods that can operate in complex, dynamic, and analytically intractable molecular environments. Tracing this trend, we can group the most recent survey papers into four categories: (i) IoBNT frameworks and applications, a field that also integrates MC and electromagnetics; (ii) theoretical and technical developments in MC, which links communication theory with MC schemes; and (iii) recent AI innovations in the MC field as summarized within the first surveyed materials on the topic.

IoBNT/MC Frameworks and Applications: Grounded in applications for the early detection and treatment of diseases in the human body, published surveys described the IoBNT architecture to fulfill a futuristic paradigm: To connect human cells to the internet [2]. The IoBNT architecture comprises a nanonetwork within the human body, in which nanobiological devices are the main components, e.g., engineered bacteria, human cells, miniaturized biosensors, and wearables. The nanonetwork is connected to wearables attached to the skin surface, which are then linked to healthcare providers via the internet; see an illustration in [4, Figures 7 and 9]. These conceptual developments and early-stage technologies embody the IoBNT in a symbiotic relationship with the collective intelligence of the body [5].

Towards this vision, existing surveys have reviewed a variety of interfaces between nanodevices and the internet, enabled by heterogeneous links that exploit the biological and electromagnetic domains, which are among the most relevant; see examples in [6] and [7], respectively. Such interfaces enable groundbreaking healthcare applications, such as reducing the detection time for bacterial infections, as illustrated in [4]. Applications also span smart agriculture and environmental monitoring, tracking the health status of animals and plants, as summarized in [6], [8], [9]. In industrial environments, nanodevices can be deployed inside underground pipes to detect corrosion and damage [8, Sec. III.b].

In the electromagnetic domain, surveys have addressed analog front-end units in the terahertz (THz) band as a possible interface between implanted nanosensors and the outside world, as in [10]. In graphene materials, the plasmonic effect observed at this frequency band is of particular applicability due to the existence of signal generators [11] and antennas [12], [13] that can be miniaturized to a few microns in length and width. Besides the THz band, recent surveys also pointed to optical bands exploiting nano-lasers and nano-antennas, megahertz (MHz) frequencies with magnetoelectric antennas based on nano-electromechanical devices, and to non-radiative techniques, either coupling-based or wave-based, such as galvanic coupling or ultrasound [14].

Furthermore, several works have summarized potential healthcare-related applications of THz networks [12], [15], including (i) ultra-precise detection and localization of diseases with nanomachines flowing in the human blood system [16], [17], and (ii) brain-computer interfaces enabled by a wireless interaction with the human brain [18] even down to single neurons [19]. Notably, however, none of the existing work on the electromagnetic side of nanonetworks identifies AI/ML as a potential tool for modeling communication or protocol design.

Recent surveys on IoBNT also highlight the self-power capabilities of nanodevices in maintaining continuous operation of nanonetworks. Micro-batteries built with micro-electromechanical and nano-electromechanical devices are powered by energy harvested from the environment or through wireless power transfer, as described in [6], [20]. Besides, due to the highly sensitive applications of IoBNT networks, especially in healthcare, surveys also note the challenges and

[2]Here, we invite the reader to follow the collage model for reading this survey, as first suggested by Julio Cortázar within the novel *Rayuela*.

risks of MC [21] and wireless channels; see [20, Sec. IV]. Likely attacks include eavesdropping, spoofing attacks, and jamming; see a full description in [20, Table 6]. These attacks exploit the limited computational capacities of nano nodes, which preclude the implementation of sophisticated protection algorithms. Mitigating strategies use fingerprinting based on channel identification (a domain well-suited for AI) or cryptographic mechanisms to avoid eavesdropping.

Theoretical and Technical Developments in MC Channel Modeling: MC channel models are a common research topic in the scientific community across different scales: (i) molecular scale as for free-diffusion or junction MC channels between cells; (ii) cellular scale, which includes cell-to-cell communication and information processing; and (iii) larger scales, which span human vessels, organs, and tissues [22]. These theoretical models provide the analytical basis for evaluating the feasibility of applications such as drug delivery, disease monitoring (e.g., cancer initiation and progression), and the deployment of body-network infrastructures [8]. The authors of [23] summarize the end-to-end channel impulse responses (CIRs) for various MC geometries at the molecular scale, irrespective of healthcare or industrial applications. MC channels include free-diffusion, advection, and chemical-degradation channels [23, Table 1]. At the tissue scale, mobility models for passive and active bio-nano machines are summarized in [24]. Theoretical work also addresses more practical problems arising in MC systems. Examples include methods for channel parameter estimation [25], modulation and demodulation [26], coding [27], and the formulation of spatial domain resources through multiple-input multiple-output (MIMO) schemes [28]. When analytical approaches become intractable, theoretical modeling is complemented by simulators that enable the numerical study of MC links, as listed in [23, Sec. V.A], [8, Sec. II.A.5]. This progression from theory to simulation is further linked to experimental validation, with recent surveys categorizing testbeds by their expected near-future technological readiness [29], [30].

Recent AI Innovations in the MC Field: A prevalent viewpoint expressed in some MC-related surveys is that data-driven detectors can complement model-based schemes that are limited to represent realistic MC channels [31], yet experimental validation remains indispensable.[3] Advancing the field, the community is beginning to embed feedforward NN architectures into DNA circuits in cells [33] and to investigate explainable methods for NN module operation [31]. However, only a few surveys, such as [31], [34], [35], address ML methods suitable for developing MC-based schemes. The work in [31] primarily focuses on performance comparisons between model- and data-driven detectors, the authors of [34] primarily consider the application layer, and the work in [35] succinctly summarizes biocompatible technologies for running NN modules.

Cross-Disciplinary Routes: Towards the interdisciplinary understanding among researchers in communication engineering, synthetic biology, and bioengineering, a hierarchy is proposed in [36] to map communication concepts to the biological behavior of cells. Unlike biologists, who study natural system interactions guided by holistic views, the engineering community is developing modular designs, i.e., defining communication functionalities in layers: application, data abstraction, molecules-data interface, molecular interactions, and their propagation. A second perspective toward interdisciplinary research lies in the recent application of information-semantic concepts to MC-engineered systems, in which goal-oriented communication may support the design of biochemical applications [37].

Summary Remarks: The variety of topics in the above-related surveys introduces a joint appeal: Scenario awareness must be part of MC deployments and dynamically adapt to real-world environments; see, for instance, the channel-noise dependency on data [23, Eqs. (73)-(76)], adaptive coding strategies to reduce inter-symbol interference (ISI) as in [27, Sec. IV.C], and the challenges to design high data-rate transmissions due to the non-stationary nature of MC noise, as discussed in [31]. A recent trend is that NN architectures are becoming relevant for channel estimation, synchronization, detection, and related tasks such as localization and disease detection, as illustrated in Fig. 1. NNs are primarily implemented outside the MC environment with digital technologies (as we discuss in Sec. III), although embedding NNs within the MC pipeline is a very appealing topic in the literature (as we consider later in Sec. IV).

Furthermore, three critical insights emerge from the recent literature as open aspects to research:

- Hardware-Driven AI: Unlike traditional AI deployments, NNs in the IoBNT must be "biocompatible" at the nanoscale. This requires a move beyond standard feedforward architectures toward low-energy modules that can be hosted within biological chemical networks.
- Explainable Artificial Intelligence architectures: There is a growing consensus that "black-box" ML models are insufficient for in-body environments. Modern research is shifting toward explainable NNs that utilize domain knowledge to provide explainable artificial intelligence (XAI), ensuring that responses are predictable and trustworthy.
- Interdisciplinary Convergence: AI has emerged as the definitive framework bridging synthetic biology and communications engineering. Recent efforts to define a common language for both disciplines enable the convergence of the disciplines toward prototyping "smart" biological nodes that function as active computational entities.

In the following sections, we provide an overview of the literature groundwork on NN architectures, MC environments, and comparative performance, establishing the technical baseline for convergence between biological and digital domains in IoBNT networks.

III. Neural Networks as Enablers of IoBNT Networks

This section details the reported NN-based solutions for the various layers in IoBNT networks. Different architectures have been explored, including feedforward NNs for channel

[3] See also [32] for a detailed explanation of ML architectures and their application in other research areas such as therapy development and nanomaterial design.

estimation, recurrent neural network (RNN) and bidirectional recurrent neural network (BiRNN)) for detection, convolutional neural networks (CNNs) and temporal convolutional neural networks (TCNs) for sequence decoding, and more recently, transformer-based and autoencoder models for end-to-end learning. These architectures have primarily been applied in supervised settings, e.g., for channel estimation and detection tasks, with fewer works exploring unsupervised feature learning and reinforcement learning, e.g., for synchronization and data fusion tasks.

To provide structure to the extensive body of related work, this section is organized into the following subsections: channel estimation, synchronization, detection, end-to-end learning mechanisms, and studies reported in the MAC and upper layers. Each subsection follows the same format: (i) Problem description for MC, where the motivation for using NN modules is discussed, followed by (ii) the MC environments where NNs have been deployed, (iii) description of reported NN architectures and their performance, (iv) illustrative examples of implemented code using NN modules, and (v) concluding remarks.

A. Channel Estimation

End-to-end MC channel estimators are essential to enable high-performance decoding at IoBNT nodes. In the literature, we find the application of NN-based estimators for various purposes, such as evaluating communication performance [38] and designing macro-scale receivers [39]. We categorize these estimators into parameter estimation and channel modeling approaches.

1) Problem Definition

The CIR in MC links is defined as the probability of finding a molecule at the receiver after its release from the emitter; see [23, Def. 2]. The CIR is given as a function of time, here denoted as $h(t)$ of given parameters $\{h_0, h_1, \ldots, h_n\}$. Channel estimation and modeling are the two main problems: the former concerns finding the set of parameters $\{h_i\}$, while the latter concerns finding the sequence $h(t)$. In many situations, estimating or developing a closed-form expression for the CIR is not feasible due to the inherent complexity of the MC channel's geometry and topology. Examples include the channel illustrated in Fig. 3,[4] where just the inclusion of a reflector introduces a non-linearity in the channel, preventing the development of a closed-form expression for $h(t)$.

Either of these two problems unfolds into a challenge due to the unknown variables that need to be estimated, such as the communication distance, the impact of the geometry of the emitters and receivers, or the fluid velocity in vessel-like media. Due to the difficulty of these problems, literature reports NN-based solutions leveraging on their capacity as universal approximators. In the following subsections, we detail various MC scenarios and the reported NN-based estimators.

2) Environments for MC Channel Estimation

In the literature, NN models for channel estimation and modeling have been applied for two main MC environments:

[4]The reflector and receiver components in the figure were generated with the assistance of ChatGPT, utilizing a Da Vinci style for the drawing. The image for the cells in Fig. 4 was generated similarly.

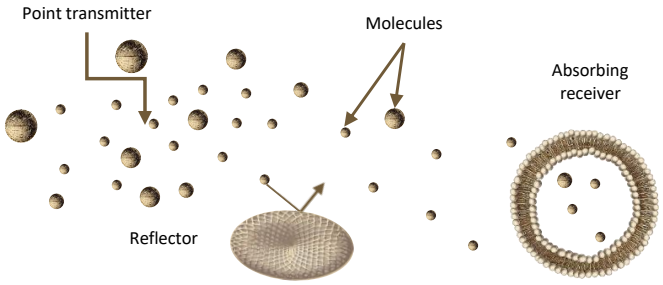


Fig. 3: Conceptual illustration of a free-diffusion channel.

for simulated free-diffusion MC channels, as in [38], [40], [41], [42], [43], [44], [45], [46], and for real-world testbeds for open air channels using sprayers as transmitters, as in [39]. The more commonplace environment of free diffusion poses challenges in evaluating the channel model for complex topologies. Examples include scenarios where multiple senders, reflectors, and absorbers are situated between the emitter and the receiver, which hinder the analytical development of closed-form solutions for the CIR; see Fig. 3. The NN-related literature considers free-diffusion environments on the (μm) scale, which include spherical reflectors and absorbers placed at arbitrary locations between a point emitter and a spherical absorbing receiver [38], [41], [42], [45].

MIMO links in MC using point transmitters and absorbing-receiver models are also gaining interest. Examples are the 2×2 MIMO scheme in free-diffusion MC channels for [40], [44], 4×4 MIMO in [46], and asymmetric MIMO channels comprising 2 emitters and 4 receivers in [43].

The literature also evaluated testbed measurements that account for realistic channel effects, including the impact of droplet size, evaporation, and unsteady flows, as in [39]. The testbed deploys an air-compressor sprayer to release ethyl alcohol molecules and an MQ-3 alcohol sensor to detect droplets from distances of up to $200\,\mathrm{cm}$.

3) NN Models for MC Channel Estimation

Various models are reported for free-diffusion channels aiming to estimate CIR parameters, as in [39], [40], [43], [44], [45], and more broadly, to estimate the CIR sequence as in [41], [42], [46]. These models implement a variety of architectures such as feed forward NN [38], [39], [40], [43], [44], [46], CNN [41], and the popular transformers [42], [43], [45].

NN-based Solutions for MC Channel Estimation: The aforementioned NN architectures are used to estimate a variety of MC channel parameters, including distances and the orientation of emitters and receivers in the channel. For instance, a feedforward NN model is developed in [39] to estimate the distance between the emitter and the receiver. The model employs a single hidden layer with a single node and is trained on a predefined set of features, which includes the peak of the received sequence and the rise time from low to high amplitude in the sequence of received molecules. Results exhibit relative errors between the estimated and actual distances in the range of 2 to $20\,\%$ at communication distances of 100 to $200\,\mathrm{cm}$.

NNs are also reported in [38], [40] to evaluate the parameters of the first-hitting time probability; see [40, Eq. (2)]. The deployment consists of single- and two- NN machines, trained on the recorded number of molecules at the receiver. The single NN is deployed through fully connected (FC) hidden layer and a total of 30 nodes. The two-machine model is proposed in [40], in which each NN estimates a separate set of CIR parameters. Both NNs are implemented with a single FC layer and 15 nodes each. The results in [40] indicate that the single-machine model performs better than the two-machine ones. These solutions achieve high accuracy with an absolute error of less than 10^{-2}.

In free-diffusion MIMO environments, NNs are trained to directly estimate the parameters of a given closed-form expression for the CIR, as indicated in [40], [43], [44]; see the model and the parameters in [40, Eqs. (2) and (3)].[5] These solutions in [40], [44] follow a two-stage approach to train the NN. In the first stage, the authors fit a CIR model to a 2×2 MIMO MC channel, using the non-linear least-squares method and samples generated via simulations. Next, the NN is trained to predict the fitted CIR using MC channel parameters such as distance, the molecule's diffusion coefficient, and receiver radius as inputs.

Continuing with MIMO channels, a more typical deployment trains NN models using received molecule counts, as investigated in [43], [46]. The work in [43] compares the performance of transformer and feedforward NN architectures for channel estimation, with the latter achieving better performance in simulations. The feedforward NN comprises one FC layer and 5 hidden layers of 10, 20, 40, 20, 10 neurons. The encoder and decoder components of the transformer implement two FC layers and the same number of hidden layers as the feedforward NN. In particular, the work in [46] estimates the distance and the rotation angle between the emitter and receiver planes of a 4×4 MIMO setup. The solution deploys a feedforward NN with 14 connected layers, where the inputs are the numbers of molecules received and the outputs are the MC distances and rotation angles.

NN-based Solutions for MC Channel Modeling: NN models are also trained to predict the CIR sequence in free-diffusion channels as explored in [41], [42]. The solution branches into two paths, one of which takes as input the red-green-blue (RGB) image of the MC environment. Point transmitters in the MC channel are depicted with dots, while absorbers, reflectors, and receivers are depicted as circles of different colors and radii. A CNN is used in [41], and a transformer (without the positional encoders) is used in [42] to extract the relevant image features. A multilayer perceptron (MLP) block is included within the architecture to normalize the sequence for the CIR. The second branch of the design in [41] employs an FC layer and an MLP block to estimate the maximum number of received molecules. As a result, the normalized mean square error (MSE) between the ground truth and the estimated CIR sequence is on the order of 10^{-2}.

[5]Closed-form CIR models can be derived for ideal channel geometries, e.g., comprising a point transmitter, a free-diffusion channel, and a spherical receiver; more realistic MC channels instead require data-driven estimation.

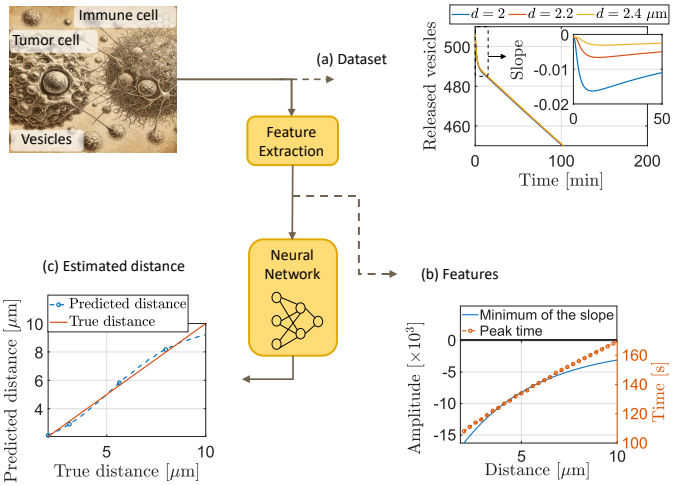


Fig. 4: Schematic for estimating the distance between cells.

4) *Illustrative Code Example to Estimate the Distance Among Cells*

To connect the preceding discussion with implementation, we present a concise NN-based distance-estimation example for a representative cell-to-cell MC scenario. This code example develops a distance estimator between immune and cancer cells based on the recorded number of vesicle molecules. Cancer and immune cells exchange vesicles in the proximity of each other, and the concentration level of vesicles recorded at the immune cell can be readily used to estimate their distance to the cancer cell; see the sketch in Fig. 4 and model details in [47]. The dependence of the concentration level on the distance parameter prevents deriving a closed-form inverse relationship; therefore, a trained NN is employed to learn this mapping.

To model the environment, we use the code provided by the authors in [47], which calculates the number of vesicles released by the immune cell; see the various curves plotted over time and at different distances in Fig. 4a. We first extract two features from the raw data to train the NN: (i) The peak amplitude of the slope of the received vesicles; and (ii) the time location of the peak (peak time), as illustrated in Fig. 4b. We implemented a feedforward NN in Matlab with a single layer and two nodes, reflecting the approximately linear relation between peak time and distance observed in Fig. 4b), which can be captured with two trainable coefficients. As depicted in Fig. 4c, the NN accurately estimates the distance to the tumor cell in the range of 2 to 10 μm with a relative error of 3.3 %.

5) *Concluding Remarks*

The use of NNs for MC channel estimation has mostly focused on feedforward NN and CNN architectures. Feedforward NNs are used to estimate specific parameters in the channel, and CNN architectures are adopted for the more ambitious goal of estimating the CIR. This subject still requires further development of NNs to target more realistic scenarios. In IoBNT contexts, NN-based channel models have not yet adequately addressed both short-range links, such as cell-to-cell communication, and long-range channels, such as those in the human circulatory system.

B. Synchronization

Successful communication among nodes relies on all participants being aware of the timing of the other entities involved; i.e., they must be synchronized. Many works on MC assume perfect synchronization by default, thereby enabling separate analysis of receiver noise sensitivity or detection parameter optimization. Yet in real-world systems, communicating nodes must explicitly synchronize to ensure successful communication. The following sections will define the problem, summarize reported approaches, and describe an example implementation.

1) Problem Definition for Synchronization

The problem of symbol synchronization concerns the question of when a symbol is considered to start at the receiver's side. In the simplest case, i.e., one transmitter and one receiver, the transmitter sends the symbols, and, according to the delay introduced by the MC channel, they arrive at the receiver. It is the receiver's challenge to determine the start of the transmitted symbol from the received sequence.

Traditionally, symbol synchronization in MC is often addressed using the maximum likelihood estimation (MLE) as an optimal method; see [48]. Due to the high computational requirements for implementing MLE, it is often not a viable option for experimental MC systems. Out of this need, low-complexity versions of MLE that rely on the use of separate signaling molecules for data transmission and synchronization were proposed in [48]. Additionally, by using two different signaling molecule types, the solution in [49] creates an artificial synchronization clock by exploiting differences in diffusion coefficients. Due to their higher diffusion coefficient, synchronization molecules arrive at the receiver before the data-signaling molecules, thereby indicating the start of transmission. A blind synchronization approach based on MLE was proposed in [50], in which a predefined sequence of molecules is transmitted before the actual data transmission to calibrate MLE.

In the synchronization approaches reported above, one problem is ignored: the variability of the MC setup. The system is assumed to be static and well-known. However, this assumption does not hold for real-world scenarios where transmitter-receiver distances, background flow, and the behavior of signaling molecules change over time. This variability triggers the need for synchronization to enable adaptation to the system's changing state. These problems are well-suited for ML-based synchronizers because they can adapt continually to fast-changing channels.

2) Relevant Environments for Synchronization

The reported ML-based solutions to synchronization constitute a first step towards incorporating more complex environments into their setup and evaluating two types of dynamic transmission scenarios. In the reported works, two extensions to the simple static MC channel were evaluated:

Mobile Setup: In many expected MC applications, the communication participants are mobile and operate within a defined space subject to environmental drift. As a result, communication may occur from varying relative positions. The synchronization strategy must, therefore, be able to adapt to a changing number of molecules being received over time.

Relay Setup: Depending on the environment, extending the communication range may require the use of relays. In this case, a single receiver will likely detect molecules transmitted by the last relay node in a drift-affected channel, along with interference from previous relays. Combined with the possible mobility of communication participants, a synchronization unit must be able to synchronize reliably in dynamically changing interference scenarios.

Still, these scenarios do not fully describe all expected application environments of MC. Synchronization strategies for MC must additionally consider more realistic setups such as multiple transmitters and receivers or multi-path propagation. Here, again, the synchronization method must be able to adjust to dynamic environments, and ML could be highly beneficial.

3) NN Models for Synchronization

As synchronization in a dynamic MC setup relies on continuous adaptation to an ever-changing environment, reinforcement learning (RL), which is based on interaction with the environment, is especially fitting. With its ability to estimate complex mathematical models [51], RL offers the possibility of solving the problem with little a priori knowledge.

Synchronization with the help of ML was first employed in [52]. An RL agent was designed for a simulated MC setup of an air-based MC testbed with a mobile receiver [53]. With rewards based on the correctness of the decoded bit, the agent learned to adjust the decoding threshold to detect a synchronization sequence. While the agent showed potential, the necessity of knowing the correct bit value during training significantly impacted the applicability of the approach in a real-world setting. The synchronizer has also been integrated into a relayed mobile setup consisting of a transmitter, a relay, and a receiver, as described in [54]. The new RL agent's reward was based on the difference between the counted number of molecules and the threshold set by the agent. The synchronizer was reported to achieve a true positive rate (TPR) of over $80\,\%$ and a false positive rate (FPR) of below $5\,\%$ for all transmitted synchronization frames. The results compared particularly well to the filter-based maximum likelihood estimation (FBMLE) synchronizer [48], which struggled to cope with the additional interference caused by the original transmission on the link between the relay and the receiver. In both RL-based synchronizers, a proximal policy optimization (PPO) agent was used. Their actor and critic networks included long short-term memory (LSTM) layers to account for past observations and actions when evaluating the current step.

Synchronization and detection via deep neural networks (DNNs) was implemented in [55]. Two setups, featuring a one-dimensional CNN and a gated recurrent unit (GRU) RNN, were evaluated for their synchronization and decoding performance, using a padded Barker code as a synchronization frame. The first setup used one NN for both synchronization and decoding, while the second setup used separate NNs for the two tasks. In the evaluation, different static signal-to-noise ratio (SNR) scenarios were considered in an unbounded free-diffusion channel. Both setups successfully synchronized the transmissions at SNRs of 45 dB or higher.

While the reported results demonstrate the usability of ML to synchronize MC systems, they only scratch the surface of the approach's advantages. Especially in more complex environments that might include multipath and multiple simultaneous transmissions, significant benefits are expected from employing an inherently adaptive ML-based synchronizer. Similar trends have been observed in wireless communication systems, where learning-based synchronization enhances robustness to channel impairments [56]. In the following section, we demonstrate how the reviewed ML approaches naturally extend to an MC setup with dynamically varying interference. We provide an illustrative example showing how such a model can be designed, trained, and evaluated in practice.

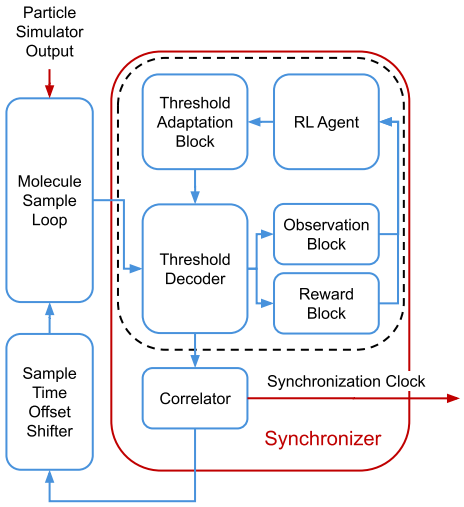
4) Illustrative Code Example to Synchronize the Receiver and Emitter Symbol Time

Following the example in [54], we implemented an RL-based synchronizer in Matlab and Simulink. We utilized a particle simulation of the media modulation (MM) testbed introduced in [57], demonstrating that the reported approach can be transferred to liquid-based closed-loop MC scenarios. In the testbed, the traditional transmitter-receiver structures are extended by adding an eraser positioned in the loop, after the receiver and before the transmitter. Switchable signaling molecules [58] are used for communication in this setup. The transmitter can turn the molecules "on"; after they pass the receiver, the eraser turns them "off" again.

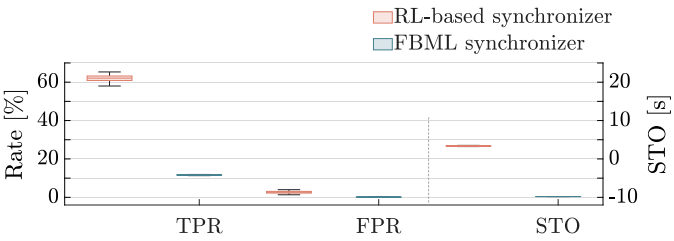
Our synchronizer follows the structure displayed in Fig. 5a. A loop through the environment consists of: (i) Sampling the current number of molecules (Molecule Sample Loop), (ii) decoding the current bit value (Threshold Decoder), (iii) having the RL agent adapt the threshold according to the observed state and the reward, and (iv) synchronizing the sampling offset if the synchronization frame was detected (Correlator and Sample Time Offset Shifter blocks). The presented system uses a PPO agent [59] with a single LSTM layer in both the actor and critic networks. We evaluated learning behavior across several network and layer sizes and found that the agent performed best with 128 cells per LSTM layer. Additionally, we performed hyperparameter tuning for the actor and critic learning rates, mini-batch size, experience horizon, entropy loss rate, discount factor, and reward scaling factor, as described in [60]. For the other parameters, we found that their influence on the agent's performance was best left at the default parameter settings. The process of setting the parameters was based on our experience from previous work [54].

We produce data for our implementation with a particle simulator of the testbed. Bits were transmitted using binary concentration shift keying (CSK) modulation with an instantaneous pulse releasing 10^3 molecules to represent bit `1` and no molecules for bit `0`. To vary the data slightly while not distorting the simulated channel behavior too much, additive white Gaussian noise was added to create an SNR of 30 dB, as described in [26]. The RL agent was then trained for 245 000 5-bit-frames in 35 000 episodes.

We evaluate our synchronization unit by comparing it to the FBMLE synchronization unit [48]. We use its synchronization bit-sequence `[11001]` as transmitted with the CSK modulation. Both synchronizers ran 100 times 1000



(a) RL-based synchronizer [52].



(b) True positive rate (TPR), false positive rate (FPR), and symbol time offset (STO) [54].

Fig. 5: System model of the RL-based synchronization unit and its performance.

5-bit-frames taken from the evaluation dataset. To assess the accuracy of the synchronizers in detecting the transmitted synchronization frames, we evaluate the TPR, the FPR, and the symbol-time offset relative to the synchronization frames for both synchronizers. The results in Fig. 5b reveal that the RL-based synchronizer achieves a higher TPR than the FBMLE synchronizer. At the same time, it also detects more synchronization frames incorrectly, resulting in a higher FPR than the FBMLE approach. Taking both detection rates together, the RL-based synchronizer performs better. Regarding the absolute value of the symbol time offset, both synchronizers accurately detect the symbol's start.

5) Concluding Remarks

The reported solutions show the potential of using ML-based approaches for synchronization in MC. Their ability to adapt to the highly dynamic environments of MC channels allows smart synchronizers to outperform other synchronization approaches in real-world settings. While existing solutions employ only RL approaches and RNN structures, research integrating diverse network architectures and attention mechanisms is rife with opportunities. Besides, as we integrate MC into the human circulatory system (HCS) for precision medicine applications, ML-based approaches offer the next step toward synchronizing workflows within the ever-changing human body environment.

C. Detectors

This section examines NN architectures for information decoding at receiver nodes, which is the most-studied topic

in the literature.[6] A key benefit of NNs is their ability as universal approximators, enabling adaptable tuning of decoder parameters in unknown environments. In line with Occam's razor,[7] NN models are designed to learn the minimum possible MC parameters of the CIR. The literature reports architectures based on feedforward NNs, RNNs, CNNs, and transformers for decoding tasks. In the following, we provide details on the detection problem definition, environments, deployed NN architectures, and an illustrative code example.

1) Problem Definition to Decode Incoming Symbols

Decoding digital transmissions refers to identifying received symbols out of an alphabet of possible ones. Similarly to wireless communications, the decoding problem is formulated based on a probabilistic description of the MC channel. MC channels are abstracted with the conditional probability $P_{\mathrm{ch}}(y_k, y_{k-1}, \dots | x_k, x_{k-1}, \dots; H)$, where H refers to the set of channel parameters, while x_k and y_k are the transmitted and received symbols in time slot k, respectively. Assuming a known probability mass function for the transmitted symbols, denoted by $P_X(x_k, x_{k-1}, \dots)$ over a finite alphabet $\mathcal{X}$ ($x_k \in \mathcal{X}$), the MAP estimate of the transmitted symbol x_k is given by $\hat{x}_k = \underset{x_k \in \mathcal{X}}{\operatorname{argmax}} P_{\mathrm{ch}}(y_k, y_{k-1}, \dots | x_k, x_{k-1}, \dots; H) \times P_X(x_k, x_{k-1}, \dots)$, see [61, Sec. II].

Solving the above problem is typically infeasible because the conditional distribution $P_{\mathrm{ch}}(y_k, y_{k-1}, \dots | x_k, x_{k-1}, \dots; H)$ and the set of parameters H are unknown. This challenge is amplified in time-variant channels, such as those in drift fluidic environments [62], and in experimental testbeds where the end-to-end model depends on complex, often unmeasurable geometries [63]. Consequently, data-driven approaches based on NNs have been proposed to learn the channel distribution directly, avoiding explicit modeling of the end-to-end MC channel.

Decoding may also require determining the optimal detection thresholds (λ_i) for CSK transmissions, which is a second problem formulation stated in the literature; see [64, Eq. (26)]. For optimal performance, the λ_i's must be adjusted based on the distance between emitter and receiver, as well as the emission pattern, which is not a trivial problem, see the formulation in [64, Eq. (13)] for the simplest case of a zero-bit memory receiver. Threshold-based detection requires solving the problem $\lambda_i = \underset{\lambda}{\operatorname{argmin}} \mathrm{BER}(\lambda)$, see [64, Eq. (26)], where BER refers to the bit error rate. This formulation is also solved through NN models due to the lack of suitable closed-form expressions.

2) Environments for NN Detectors

NN models, as detectors, are trained in MC- free diffusion environments, where multiple geometries are considered in the literature. The training sequence is constructed in MC channels where the emitter is a point transmitter and the receiver is an absorbing sphere [64], [65], [66], [67], [68] or a ligand receptor [69], [70]. More complex environments follow multi-hop links as in [71], 5×5 MIMO channels with ligand receptors in [70], and a 8×1 multiple-input single-output (MISO) channel as described in [72]. Mobility of the transmitters and receivers is also modeled in [73], [74], [75], [76], which eventually distorts the received sequence. Vessel-like channels, which have been less studied, were introduced in [77]. The communication link is established inside a pipe, where the emitter is fixed in place with a cylindrical geometry, while the receiver moves across the pipe's cross-section and oscillates along the flow direction.

The randomness of the MC channels is primarily modeled using the Poisson distribution, as in [61], [64], [65], [73], [74], [78], [79], and it is less frequently modeled with the Gaussian distribution as in [72], [77], [80]. More realistic MC models also incorporate the degradation of molecules due to chemical reactions, as evaluated in [66], [73], [74].

Received training sequences have been recorded from experimental testbeds in vessel-like channels. Examples include binary emissions performed with acid and base signals in water [61], [80], [81], [82], [83], [84], [85], saline solutions and water [86], and using magnetic nanoparticles as in [87], [88]. Binary transmissions are performed using *E. Coli* bacteria as generators, as reported in the *in vivo* testbed in [89], [90]. *E. Coli* bacteria are stimulated with light to release protons and increase the medium's pH level, and a sensor measures the pH levels, serving as the receiver [89]. As another example, binary transmission is performed by releasing flagellated bacteria, which are guided by a magnetic field towards a spherical receiver, as described in [90]. The receiver, which was initially filled with luminescent non-motile bacteria, begins collecting the quorum sensing (QS) molecules released by the arriving bacteria and produces light in response. The observed intensity and the number of collected bacteria are used to subsequently decode the transmitted sequence.

To summarize, the communication methods outlined above use pulse-based modulation, i.e., on-off keying (OOK). The one bits are encoded with the number of released molecules or bacteria, and the zero bits are encoded by their absence. Additionally, emissions occur with bit durations in the millisecond (ms) range and the channel noise varies significantly from a considerable level ($\mathrm{SNR} = 0\,\mathrm{dB}$) to a nearly negligible one ($\mathrm{SNR} = 60\,\mathrm{dB}$).

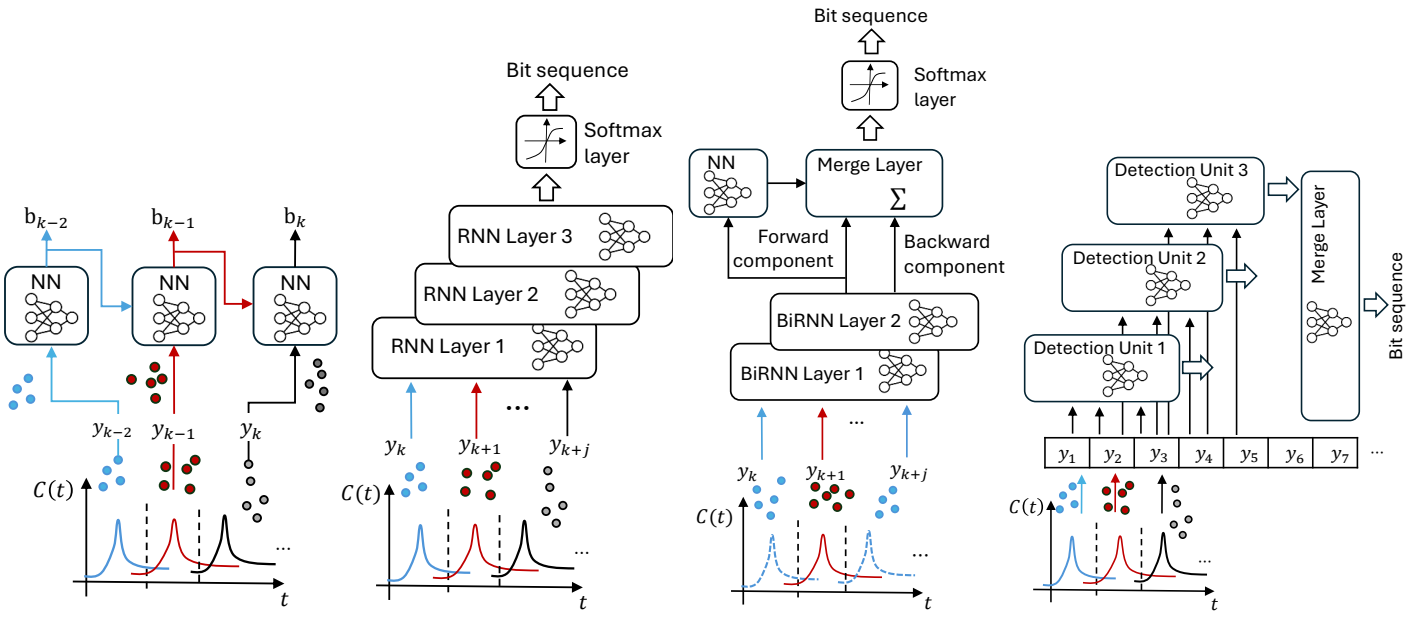
3) NN Architectures to Detect Incoming Symbols

Reported architectures for detection in MC include feedforward NNs [61], which exploit prior knowledge such as channel memory; CNNs [31], offering robustness to channel delays; and BiRNNs [61], which effectively mitigate ISI by processing both past and future samples when decoding the current symbol. More recently, attention-based models [91] have been introduced to capture long-range temporal dependencies and further improve decoding performance in channels with extended memory.

These architectures are reported to operate either on a symbol-by-symbol or on a sequence basis. In symbol-by-symbol mode, the NN is trained with samples corresponding to a single symbol only, like in [64], [67], [69], [77], [78], [81], [92] (simplest detection mode). More robustly, in sequence detection, the NN is trained on several symbols. In this way, the detection of the current symbol leverages observations of

[6]The total number of references related to detection tasks is 26 % of the surveyed literature.

[7]Occam's razor is the principle that, among competing element sets, the "simplest" one should be preferred, i.e., the set with the smallest number of elements.



(a) Cascade connection of feedforward NNs. (b) Cascade connection of RNNs. (c) Cascade connection of BiRNNs. (d) Sliding window detector.

Fig. 6: Feedforward and recurrent NN architectures as sequence decoders. The RNN modules implement LSTM cells; see [61], [81].

past and future received ones, as in [64], [65], [74], [83], [86], [92]. The architectures that follow these two approaches are described next.

Symbol-by-Symbol Detection Mode: In this mode, feedforward NNs solve the maximum a posteriori (MAP) formulation stated in the previous section, as described in [66], [67], [69], [73], [81], [85], [89], [90], [92]. RNNs and CNNs architectures are also reported, as in [89] and [72], respectively. Besides, symbol-by-symbol detectors can be trained more efficiently by using distinctive features of the input sequence. Features are evaluated based on the concentration difference of received molecules [61], [66], [73], [81], [89] or by fitting a polynomial to the received concentration trace and using its coefficients as feature inputs, see [89]. Other examples extract features from the received sequence using CNN architectures comprising one [81], [89], [90] and three [77], [87], [88] hidden layers.

Sequence Detection Mode: In this mode, the reported NN architectures are more diverse; not only feedforward NNs models are used [64], [65], [74], [78], but also CNNs [70], RNN [70], [81], [86], and BiRNNs [61] encompassing LSTM cells. A cascade connection of the feedforward NN architecture is reported in [64] operating as a threshold decoder. This solution aims to determine the optimal threshold (λ_i) that minimizes the BER metric, as stated in the section above. As depicted in Fig. 6a, decoding the current bit (b_k) accounts for the previously decoded ones (b_{k-1}, b_{k-2}), allowing the NN to inherently learn the ISI effect in the MC channel.

The long-term dependencies in the input sequence are more efficiently learned using LSTM cells within RNN architectures, as described in [61], [81]. The schematic of these decoders is depicted in Fig. 6b, where three layers are connected in cascade, each layer is of length 40 and trained with 120-bit sequences; see [81, Fig. 1 b)]. Besides, with LSTM networks, bidirectional architectures can be more effectively deployed to exploit the correlation of past and future samples for decoding; see [81, Fig. 1 c)]. Past emissions cause ISI; therefore, knowledge of the previously transmitted bits can be used to cancel their effect on decoding the current bit. Meanwhile, as the current emission leaks into the next, the future samples also carry information that helps decode the current one. Exploiting input data of past and future samples, the implemented BiRNN model in [81], [92] reduces the impact of ISI and is reported as being more computationally efficient than the Viterbi decoder for long memory channels.

The above BiRNN architecture is enhanced by adding a learning mechanism that merges the forward and backward pathways, as reported in [83, Fig. 1b]. As depicted in Fig. 6c, a feedforward NN adjusts the coefficients of the weighted sum for the merging layer. The model is trained using the Adam optimization algorithm (see [93]) and achieves a lower BER (one order of magnitude lower) than detectors using feedforward NNs or CNN architectures.

The above architectures can also be integrated into the sliding window architecture to further improve performance [61]; see the schematic in Fig. 6d. Three detection units decode overlapping symbols within a sliding window, aiming to detect larger correlation lags in the input sequence. For instance, in deciding the bit sequence within the symbol y_3 in Fig. 6d, not only the neighbor samples y_2 and y_4 are processed (within the detection unit 2), but also y_1 and y_5 (within the detection units 1 and 3, respectively). In this way, the Merge Layer block is fed with a more extensive sequence when compared to the same block in Fig. 6c. In this solution, the detection units are implemented using BiRNNs and the Merge Layer block evaluates the average of the RNNs outputs; see [61, Eq. (10)].

Attention Models in the Loop: Attention models are increasingly referenced for improving NN architectures in MC channels; see recent examples in [68], [71], [75], [76], [80].

As their main feature, these models estimate the relevance of data within the input sequence, which improves subsequent decoding. Although previous architectures, such as BiRNN, inherently include this feature, they are typically limited to short-range dependencies, whereas attention models can span longer ranges within the input sequence; see [94]. Seeking to reduce complexity, the encoder component of the transformer is implemented with a single self-attention unit, and the decoder component comprises only the NN unit, as described in [71], [76, Fig. 2]. Additionally, the authors in [68] further investigate reducing the transformer's hyperparameters (such as the encoding vector length and input size) while maintaining standard operation. As an alternative, the encoder component of the transformer can be replaced with a three-layer CNN module connected to the decoder component's input. Moreover, to facilitate a more robust architecture, the sliding window scheme in Fig. 6d has been extended by implementing the detection units using a transformer, as in [76]. Next, to illustrate the operation of NNs in more detail, we present a short code example that highlights how the BiRNN architecture is designed, trained, and tested in a practical MC setting.

4) Illustrative Code Example for Symbol Detection

This section demonstrates the performance of the sliding BiRNN architecture using samples from the experimental testbed illustrated in Fig. 7a. The testbed sets a communication link over the distance of $1\,\mathrm{m}$ between the emitter sprayer and the receiver sensor. Using ethanol molecules as carriers, OOK emissions are performed with binary pulses of $T_b = 4\,\mathrm{s}$ duration each; see further details in [53, Sec. II]. We use the recorded received pulse as the expected sequence for the emission of ones, and we model the randomness of the received sequence using the Poisson channel model; see [23, Eq. (87)]. Besides, we manually (in the code) added synthetic noise molecules of concentration $10\,\mathrm{mg/L}$, yielding an SNR of $27.5\,\mathrm{dB}$. At the receiver, the observed pulse sequence is sampled with the synchronization signal, which is assumed to be known.

Due to the channel memory, consecutive emissions cause ISI, hindering the use of low-complexity schemes, such as the threshold decoder. For the decoder, we train the sliding BiRNN scheme reported in [61], [82] as a sequence detector. With the measured pulse duration from the experimental testbed ($28.6\,\mathrm{s}$ in total), we evaluate the channel memory as $L = \lfloor \frac{28.6\,\mathrm{s}}{T_b} \rfloor = 7$, which also defines the number of hidden states of the BiRNN model. For training, we use the Adam algorithm [93] with a learning rate of 10^{-3}, 10 epochs, a batch size of 10 units, and an emission of 10^5 bits.

The selected hyperparameters ensure stable learning, with the training loss decreasing over epochs and the bit duration, as illustrated in Fig. 7 b). Specifically, we observe stability after the second epoch, and losses decrease as the bit duration increases. The resulting BER falls within the range 10^{-2} to 2×10^{-6}, where more errors occur for lower bit durations, i.e., $T_b < 2\,\mathrm{s}$. Besides, as the peak of the received pulses lasts for around $4\,\mathrm{s}$, the resulting BER dramatically decreases when transmissions are performed with the bit duration of $5\,\mathrm{s}$ or more, i.e., when ISI becomes negligible.

5) Concluding Remarks

NN models have proven to successfully cope with the challenging nature of MC channels as detectors, achieving BERs of less than 10^{-4} in testbed environments. The variety of environments and NN architectures is rich in the literature: Feedforward, recurrent neural networks, and attention models have been researched to decode sequences. These architectures have been tested in diffusion and drift channels, using molecules and bacteria as information carriers. However, upon reviewing the literature, little is found regarding realistic environments for precision medicine applications; the considered MC channels have mostly simplified geometries, such as free-diffusion point-to-point links. In future work, diffusion models can be extended to target cell-to-cell communications, including extracellular and intracellular channel models. Mobility models can be enhanced to replicate the complex communication environment in human vessels, presenting valuable opportunities for further NN-model research. Moreover, integrating these decoders with MC channel estimators and noise suppressors, as in [95], will lead to improved performance. Future studies should aim to balance the added architectural complexity caused by this integration with its impact on performance.

D. Autoencoders

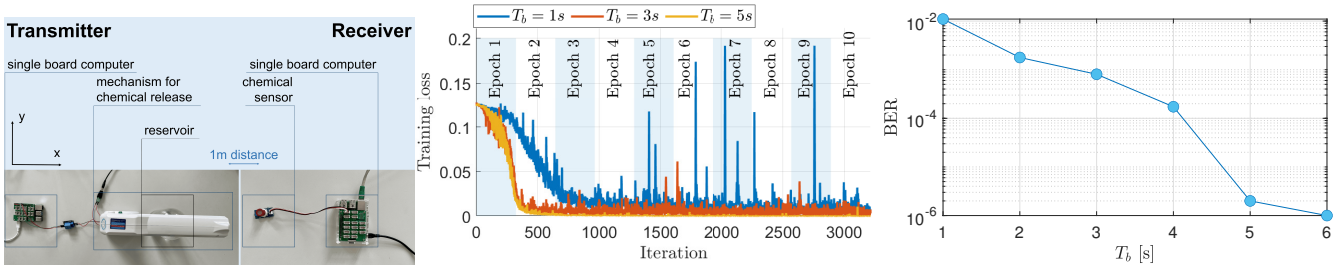
End-to-end learning using autoencoder (AEC) is an innovative approach in MC, which, on the one hand, mitigates the challenges associated with MC channel modeling, and on the other hand, optimizes the entire MC system, including both the transmitter and receiver [96], [97]. Traditional MC systems typically divide the transceiver chain into distinct blocks such as modulation, channel estimation, synchronization, equalization, and demodulation. However, greater benefits can be achieved from a design that integrates the entire communication system (end-to-end), and AEC serves as a prime architecture to optimize the entire process [98].

An AEC consists of two NNs, known as the encoder and decoder, which are trained together in an unsupervised fashion; see a representation in Fig. 8. The encoder and decoder replace the transmitter and receiver components of the communication system, and the entire communication pathway is optimized end-to-end [99, Sec. 1.3.2 pp. 11]. This section summarizes the reported literature on AECs for MC, defines the end-to-end learning problem in MC, describes the deployed NN architectures, and finally, provides an illustrative code example.

1) Problem Definition for Autoencoders

The AEC architecture comprises DNNs for the encoder and decoder, with separate trainable parameters. The problem is to find these trainable parameters so that the information transmitted by the emitter is restored at the receiver. Within the MC environment, the encoder generates an optimal number of emitted molecules (referred to as symbol generation or constellation design), ensuring that it can be accurately detected at the receiver side with minimal error, despite the stochastic nature and variations of the molecular channel. Therefore, the ultimate goal is to train the AEC to minimize a desired loss function; see the corresponding block (rightmost) in Fig. 8.

As for the loss function, an AEC adopts the cross-entropy as an effective metric for the end-to-end learning of systems. This metric measures the difference between the predicted



(a) Experimental testbed; adapted from [54]. (b) Training loss. (c) Resulting BER.

Fig. 7: Experimental testbed used for MC and measurements, as developed in [53].

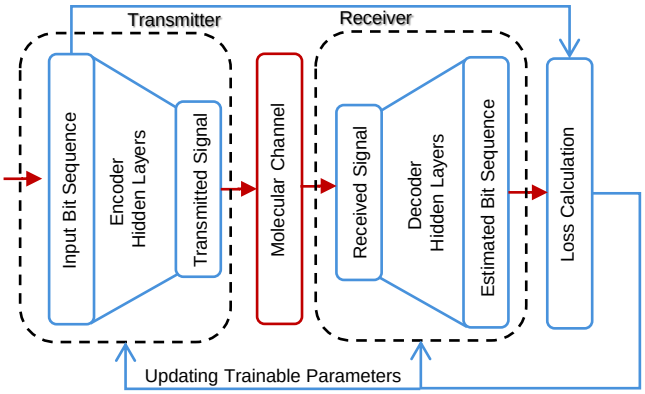


Fig. 8: AEC architecture for end-to-end learning of MC.

and the actual probability distributions of the transmitted information. For symbol-based transmission, where data is encoded as individual symbols from a predefined alphabet, categorical cross-entropy is appropriate [96]. In contrast, binary cross-entropy is more effective for block-based transmission, which processes data in the form of larger binary sequences, as reported in [100].

2) Environments for Autoencoders

Advection-diffusion channels are the main type of MC channels considered in the literature for studying AECs. In this scenario, a trained point transmitter encodes information bits into molecular concentrations and controls their release accordingly. The molecules are detected by a passive receiver, which decodes the signal and extracts the information using its trained NN, as reported in [96], [97], [100], [101].

3) NN Models for Autoencoders

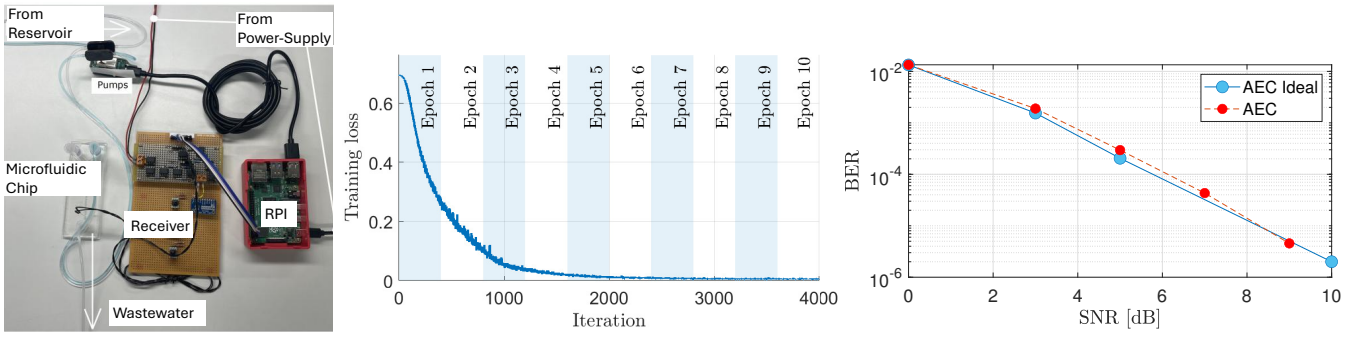
The AEC model must first undergo training before being deployed in a practical MC system. The literature presents several approaches for training AECs to facilitate end-to-end learning in MC environments, including the following.

Model-Assumed Training: Model-assumed training is an approach in which both the transmitter and receiver are trained using a predefined channel model. In MC, this involves using the estimated CIR from the testbed, implemented as a fixed, non-trainable convolutional layer between the encoder and the decoder of the AEC. An example is given in [97], where the CIR is obtained from an experimental salinity-based testbed as reported in [102]. This approach assumes that if the model from the testbed remains stable during runtime, with minimal variations, the transmitter will identify effective strategies for encoding input during training. Then, in the online phase, when actual transmission occurs in the experimental system, the receiver can be retrained to adapt to the slightly changed channel conditions. At the same time, the transmitter will maintain its previously established strategies. Thus, if the assumed model in the training phase is sufficiently accurate, the trained AEC will also work appropriately for transmission over the experimental channel.

The solution reported in [97] uses two CNNs as its architecture, each serving as an encoder and a decoder. The encoder consists of three convolutional layers, with 16, 32, and 1 filters, respectively, each using a kernel size of 3 elements. Each convolutional layer is followed by batch normalization and an rectified linear unit (ReLU) activation function. A final normalization layer ensures a controlled number of transmitted molecules. The decoder begins with a convolutional layer featuring 16 filters and a kernel size proportional to the channel memory, effectively mitigating ISI. This is followed by an adaptive average pooling layer, a fully connected linear layer, and another convolutional layer, all of which, except the last layer, incorporate batch normalization and ReLU activation. The final layer employs a sigmoid activation function. With a differentiable channel model, the AEC is trained using backpropagation.

Training via Data-Driven Channel Identification: The AEC can also be trained on a data-driven molecular channel representation, as proposed in [100]. The training procedure involves three steps: (i) Modeling the MC channel using an RNN, (ii) training the emitter and receiver components of the AEC, and (iii) fine-tuning the trained model. A data-driven ML method is used in the first step to obtain a differentiable representation of molecular channels. Specifically, an auto-regressive exogenous (ARX) model is employed as the channel representation, implemented as an infinite impulse response (IIR) filter using a trainable RNN. This formulation preserves differentiability for end-to-end optimization while more effectively capturing long-term temporal dependencies in molecular propagation than feedforward models.

In the second step, both the encoder and decoder parts of the AEC are jointly trained using backpropagation, while the RNN representing the channel remains fixed. Lastly, the AEC undergoes fine-tuning to address mismatches between the approximation of the channel model and the real model. The decoder parameters are adjusted using transmissions over the



(a) Experimental testbed. (b) Training loss for the AEC. (c) Resulting BER for the AEC when symbol duration is 300 ms.

Fig. 9: Experimental testbed developed in [102] and the results for the trained AEC in [100].

real channel, while the encoder parameters remain unchanged.

Model-Free Training: In this mode, the AEC is trained without relying on any model (analytic or data-driven). To illustrate the methodology, we consider the deep reinforcement learning (DRL) system with fully connected NNs, as proposed in [96], where information is transmitted through a stochastic, unknown channel. During each training iteration, the decoder is optimized while keeping the encoder parameters fixed, followed by optimizing the encoder while keeping the decoder parameters fixed. This iterative process enhances the overall system performance. Updating the decoder parameters is straightforward as it is a supervised task, and does not require backpropagation through the channel. However, updating the encoder relies on backpropagation through the channel, which requires a channel model to be available. To circumvent this challenge, the authors in [96] employ an RL model, treating the transmitter as an agent that receives the loss calculated at the receiver via an ideal separate feedback channel. Additionally, [103] incorporates feature-wise linear modulation (FiLM) layers into the AEC architecture to enable online adaptation during deployment, allowing the model to adjust to channel conditions that differ from those observed during training without retraining.

Transmitter-Exclusive Training: Transmitter-exclusive approaches perform the training on the transmitter side only. This training mode is well-suited for the IoBNT scenario, where the transmitter is usually easily accessible, as it is located outside the body, and the receiver is of low complexity and located inside, as specified in [101], [104]. The proposed asymmetric autoencoder (AAEC) in [101], [104] employs a CNN encoder for binary transmission with adjustable concentration levels and a threshold-based decoder realized as a single convolutional layer. This approach seeks to mitigate the impact of residual molecules from prior transmissions by encoding symbols to counteract lingering interference, thereby reducing ISI.

Moreover, a DNN-based approach for optimizing the number of molecules released by each transmitter in a mobile molecular MIMO system is presented in [105]. While AEC is not utilized in the proposed structure, this DNN-based method can be integrated into transmitter-exclusive training for MC. The optimization is performed based on different transmitter-receiver distances to minimize the inter-link interference (ILI), thereby reducing the BER. The DNN is trained with constraints on the molecule release count to ensure compliance with lower and upper bounds. The results demonstrate that the DNN-based approach outperforms the genetic algorithm (GA)-based approach, achieving a lower average BER and significantly reduced computation time. Additionally, the DNN-based optimization achieves a BER comparable to that of an exhaustive search while significantly reducing computational time.

Adaptive modulation and coding are other critical tasks to dynamically adjust transmission parameters in response to changing channel conditions. Adaptive modulation is essential for maintaining reliable and efficient communication in MC, where factors such as diffusion and advection strongly influence the channel. In this context, an RL-module has been proposed to optimize real-time transmission parameters, such as the modulation order and symbol duration, [106]. A gateway device connected to the RL model estimates channel conditions from heart rate data by leveraging a digital twin of the human circulatory system. This solution improves the achievable raw bit rate and error performance. Additionally, to address the limited capabilities at the nanoscale, this architecture alleviates the computational burden on resource-constrained nanodevices by offloading complexity to an external gateway. To complement the above review, we next provide a brief illustrative example, adapted from [100], that demonstrates how this architecture is designed, trained, and evaluated in practice.

4) Illustrative Example for Autoencoders

This section complements the above architectures by illustrating the operation of the AEC from [100] in the salinity-based testbed developed in [102]. In this testbed, information bits are mapped to salinity levels and detected based on the corresponding conductivity in water, for which a closed-form expression cannot be derived due to the lack of an accurate model of the transmitter and receiver geometries. Fig. 9a depicts the setup for a $3.6\,\mathrm{cm}$ transmitter–receiver distance. Real microfluidic channel measurements are used to identify channels and train an RNN.

The AEC employs three convolutional layers with batch normalization and ReLU activation at the encoder, while the decoder uses a convolutional layer acting as a linear equalizer to mitigate ISI and noise. Two pooling layers (average and max) and a fully connected layer downsample features to

match the transmitted bit sequence, followed by a sigmoid output for soft-input decoding. Training is simulation-based over 2000 epochs with a batch size of 40 and 100 bit sequences (Fig. 9b). Each emission comprises encoder-defined binary pulses lasting $500\,\mathrm{ms}$. The Adam optimizer is used with an initial learning rate of 0.009 and a decay of 0.99 every 1000 iterations. The architectural parameters, such as the layer type and number, as well as the number of neurons per layer, were empirically tuned based on observed training performance and prior experience, with domain knowledge guiding design choices. For instance, convolutional layers with kernel sizes matching the channel memory were adopted at the receiver side. Training hyperparameters, including learning rate, batch size, and optimizer type, were optimized through a grid search to achieve stable convergence and minimize validation loss, as described in [102]. As a result, the obtained BER ranges from 9×10^{-2} to 9×10^{-6} when the SNR is in the interval 0 to $10\,\mathrm{dB}$. These results attain the ideal case of full MC channel knowledge; labeled as 'AEC Ideal' in Fig. 9c.

5) Concluding Remarks

AEC can inherently enhance the performance of MC by jointly optimizing the transmitter and receiver operations. Besides, this architecture effectively addresses the complexities of dynamic and stochastic MC channels. However, several practical challenges remain unresolved. For example, real-world molecular channels are time-varying, requiring continuous fine-tuning of NN parameters to adapt to unseen conditions. How to efficiently update the transmitter parameters during deployment without incurring high computational or energy costs remains an open question for future research. Additionally, deploying resource-intensive NNs on small, biocompatible, and resource-constrained devices poses significant technical hurdles.

E. Higher Layers

The works summarized thus far mostly focus on the PHY layer and target point-to-point communication links. Fewer studies address higher layers, such as resource allocation (MAC layer) and localization (application layer).[8] This section provides an overview of the reported research on integrating NN architectures into the higher layers of IoBNT networks.

1) Resource Allocation

Resource allocation problems arise when multiple users attempt to communicate using the same resource. Such a scenario is examined in free-diffusion channels in [107], which investigates a transmission policy that minimizes the BER. The scenario considers an arbitrary number of transmitters placed at random locations in a free-diffusion channel, and restricted to a maximum number of released molecules.

The transmission policy is devised by a feedforward NN with three hidden layers, each containing twice as many neurons as there are transmitters, except for the first layer, which has two additional neurons, as indicated in [107]. The NN is trained using the distances from the transmitters to the receiver as inputs. The model outputs the number of molecules released by each transmitter. As activation functions, the first layer uses the hyperbolic tangent sigmoid, the "purelin" (linear) in the second hidden layer, and the ReLU for the third hidden layer. This architecture achieves a BER on the order of magnitude 10^{-3} with three to five transmitters positioned between 12.5 to $14.5\,\mu\mathrm{m}$ from the receiver. The network operates with a symbol time of $100\,\mathrm{ms}$, a total of 6×10^4 released molecules, and a noisy source modeled as a Gaussian distribution with variance 1000.

2) Localization

Localizing diseases and self-localizing nanosensors are among the most anticipated applications of IoBNT networks in precision medicine. Scenarios have been developed within the dynamic environment of blood vessels as in [108], [109], [110], [111], [112], [113], [114] and in the less restrictive case of free-diffusion environments as in [115], [116].

In blood vessels, prior work assumes that blood flow passively drives existing nanosensors. In the HCS environment, the self-localization capabilities of nanosensors are particularly challenging to develop due to the absence of a reference system. A reference system for self-localizing nanosensors within the bloodstream can be anchored to the concentrations of neighboring nanosensors and their travel times, as proposed in [108]. This work assumes that nanosensors contain an internal counter and an external device that resets the counter when it travels through the heart. Then, as the nanosensor's traveling time increases as it flows through the blood vessels, the recorded traveling time per nanosensor can be used to distinguish shorter (central body) from longer (lower body) paths, thereby self-distinguishing the body region. Additionally, as the concentration of nanosensors depends on the specific vessel path (see the Markov model formulation in [110], [117]), it can serve as a second metric for self-localization. As such, a feedforward NN can be trained to distinguish the traveling path of nanosensors based on these two metrics, i.e., traveling time and concentration level, as proposed in [109]. The NN is trained on data generated by the BloodVoyagerS (BVS) simulator [118], achieving approximately $85\,\%$ positive predictions. Localization methods also rely on establishing a communication link among nanonodes, as described in [112]. This research assumes transmitter nodes with fixed spatial positions within the human cardiovascular system and a receiver node, which is intended to be localized. Using the transformer architecture, the prediction accuracy for receiver coordinates ranges from 78 to $96\,\%$ [112], depending on blood flow velocity and the transformer's model capacity, which is determined by its computational complexity.

A localization method using graph neural networks (GNNs) is also reported in [113], where the HCS is modeled as a graph using the BVS representation with 94 regions. The proposed model relies on heterogeneous graph transformerss (HGTs) to propagate information between region and anchor nodes, using a configuration with 64 hidden channels, 8 attention heads, 3 HGT layers, and 4 convolutional layers. Compared with the feedforward NN baseline in [108], [119], the GNN improves point localization accuracy by more than $30\,\%$ in the reported $18\,\mathrm{min}$ runtime scenario and extends localization coverage to the entire bloodstream.

Localization of nanonodes is also performed in short-range free-diffusion MC links, as developed in [115] for security applications and in [116] for tracking a mobile emitter. The

[8]Contributions related to higher layers represent $7\,\%$ of the surveyed literature.

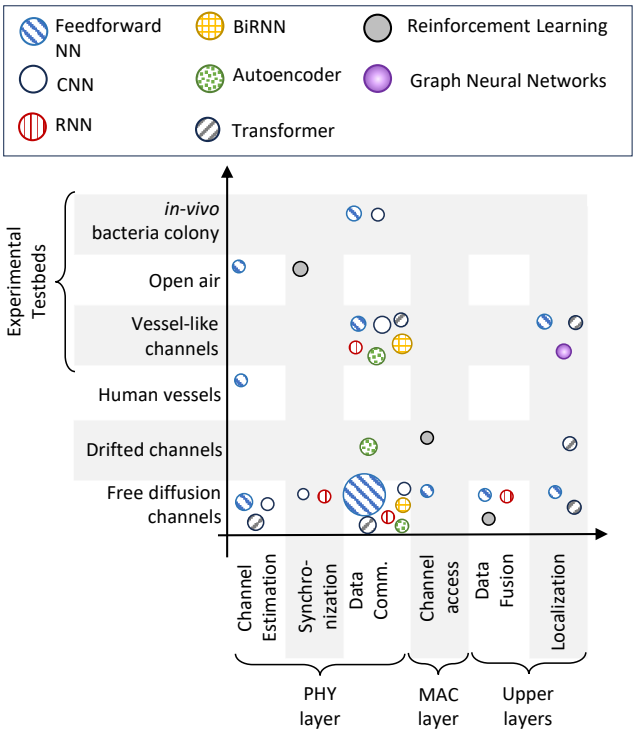


Fig. 10: Bubble plot of reported NN architectures.

feedforward NN model in [115] is developed for a 2D free-diffusion environment and localizes an eavesdropper node with an error of less than $4\,\mu\text{m}$. The solution in [116] develops a transformer architecture that tracks the position of a mobile emitter with precision exceeding 80 %.

3) Data Fusion

Reported research targets data fusion techniques in tasks related to detecting potential abnormalities [109], [120], [121], [122]. Applications include health-condition monitoring, environmental sensing, and the detection of toxic agents. Commonly implemented fusion rules are the `OR`, `AND`, or `MAJORITY` rules; see for instance [109, Sec. IV B]. However, these rules are complex to apply in practice, as they require evaluating channel-dependent parameters such as noise and interference levels, which are typically unknown. Overcoming this impediment, NNs are trained and deployed to effectively fuse data from various sensor nodes, utilizing feedforward NNs in [120] and RNN in [121]. These two architectures are developed for free-diffusion channels, achieving a detection probability greater than 0.9 and a false alarm probability of 5×10^{-2}. Furthermore, the robustness of data fusion techniques is investigated in [122], where an RL agent is trained to optimize the observation window for collecting samples at the fusion node. The RL agent adjusts the time slot duration to ensure a detection probability within a confidence interval and to minimize noise levels. This work reports the fusion of binary emissions from 100 nodes in free-diffusion channels, utilizing the `OR` rule at the fusion node.

F. Concluding Remarks and Outlook

Within the reported literature, the deployment of NNs follows two common practices: (i) The most popular models are feedforward NNs, and (ii) deployment focuses primarily on decoding tasks in point-to-point links. We identify these two characteristics from the comparative summary provided in Fig. 10 and the consolidated overview in Table I. In this figure, the size of each bubble represents the number of published works, arranged by communication layer (horizontal axis) and by MC scenario (vertical axis).[9] Table I reports key MC-related parameters that describe per MC geometry and NN architecture. This tabular overview also highlights the diversity of experimental and simulation setups used in the surveyed studies.

From Fig. 10 and Table I, we also identify several research gaps that deserve further investigation: (i) the lack of NN-based solutions for multipoint-to-multipoint links; (ii) the limited attention given to upper-layer functionalities; and (iii) the need for additional experimental validation to support the integration of NN-based schemes.[10] Furthermore, most reported NN models are trained offline and used solely for inference, while online and adaptive training strategies remain largely unexplored.

Another key aspect to consider is the feasibility of integrating the reported NN-based solutions into nanodevices, which remains open for further analysis. As nanodevices are expected to operate with limited resources, it is crucial to examine the trade-off between performance and complexity across tasks, including channel estimation, synchronization, detection, and upper-layer applications such as data fusion and localization. Following the literature, the number of parameters reported across methods ranges from below 10 to several thousand, see Table II. The least demanding architectures are feedforward and recurrent NNs and are used primarily for tasks related to data communication. The most complex architectures are reported using CNNs and reinforcement learning methods employed for synchronization and data communication.

To shed more light on this topic, we examine a variety of architectures for the data communication task. We evaluated the trade-off between detection performance and complexity across various NN architectures, and present a comparative summary in Fig. 11. This figure illustrates the performance of the various architectures, evaluated in terms of the BER as a function of the transmission rate (bit time T_b).[11] In this evaluation, all architectures were implemented at a comparable complexity, following the BiRNN reference design, and Transformer models were excluded because they performed poorly in our simulations when constrained to the same number of parameters.

The comparison in Fig. 11 confirms an expected result: The BiRNN achieves similar performance with fewer learnable parameters than the other architectures, effectively captur-

[9] As a reference, the size of the bubbles in Fig. 10 reflects 10 publications for the largest bubble, and one publication for the smaller ones. The generation of this plot is accessible in https://github.com/tkn-tub/NN_molecular_communications/tree/main/Tables.

[10] Several physical parameters used to model MC channels, such as receiver radius, diffusion coefficients, and communication ranges, are often selected for analytical or simulation convenience rather than derived from specific experimental technologies. Establishing experimentally grounded parameter values remains an important direction for future developments in the field.

[11] The BiRNN consists of five LSTM cells in both the forward and backward directions, to account for channel memory.

TABLE I: Summary of NN architectures, applications, and parameters related to the MC channels and communication.

			End-to-end channel-related parameters			Communication-related parameters					
MC Geometry	NN architecture	Application	Communication range	Receiver radius	D [$\mathrm{nm}^2/\mathrm{ns}$]	Released molecules	Symbol duration	SNR [dB]	Performance metrics		Ref.
Point transmitter-Free diffusion-Spherical absorbing receiver	Feedforward NN	Channel estimation	2 to 10 µm	4 µm	33×10^{-4}, 66×10^{-4}	200			Error	3.3 %	Sec. III-A4
			2 to 11 µm	3 to 7 µm	5×10^{-1}, 10^{-1}	3×10^3			RMSE	≤ 0.3	[40]
			2 to 11 µm	4 to 10 µm						≤ 1	[38]
			4 to 12 µm	3, 4, 5 µm		10^6				$\leq 10^{-2}$	[43]
			8 to 10 µm	4 µm	7.9×10^{-1}	3×10^3					[46]
		Data communication	500 nm	45 nm	4.2×10^{-1}		270 ms	−5 to 80	BER	$\geq 2\times10^{-2}$	[78]
							450 ms	−5 to 35		$\geq 2\times10^{-5}$	
			5 µm	50 nm	1.01	4×10^4 to 4.5×10^4	100 ms	30 to 56		$\geq 10^{-4}$	[73], [74], [66]
			14 µm	1 µm	6×10^{-2}	10^3, 2×10^3	3 s	2 to 20		$\geq 7\times10^{-6}$	[67]
		Localization	10 µm	4 µm	7.9×10^{-1}	10^4	2 and 5 s		Error	< 50 %	[115]
		Data fusion	4 to 9 µm	4 µm	5×10^{-1}, 7.9×10^{-1}		70 ms	2 to 5	P_d	>0.94	[120]
									P_fa	$\leq 7.5\times10^{-2}$	
	RNN		4 to 9 µm	4 µm			70 ms	2 to 5	P_d	>0.96	
									P_fa	$\leq 7.5\times10^{-2}$	
	BiRNN	Channel estimation	4 to 12 µm	3, 4, 5 µm	5×10^{-1}, 10^{-1}				RMSE	$\leq 10^{-2}$	[43]
		Data communication	1 to 50 µm		1	10^4 to 5×10^4	10 ms	0 to 60	BER	$\geq 10^{-5}$	[65]
	Transformer		7 to 12 µm	5 µm	5	6×10^3 to 18×10^3	35, 100 ms	0 to 40		$\geq 4\times10^{-3}$	[76], [71]
		Channel estimation	4 to 12 µm	3, 4, 5 µm	5×10^{-1}, 10^{-1}	3×10^3			RMSE	$\leq 10^{-2}$	[43]
		Localization	meter–scale						Accuracy	≤ 39 %	[114]
Drifted channels		Data communication	10 µm	1.5 µm	0.79	4×10^3	200 ms	10 to 40	BER	$\geq 4\times10^{-3}$	[68]
	Autoencoder		38 mm		1.24×10^5		100 to 600 ms	0 to 10		$\geq 2\times10^{-6}$	[100]

ing interrelations among samples and improving decoding. Moreover, the figure also displays an inferred approximation of orders of magnitude for performance–complexity trade-offs: NN architectures with hundreds of learnable parameters are generally required in MC channels with low ISI, i.e., channel memory of two to five samples. A similar analysis is still pending for other tasks such as channel estimation, synchronization, or localization. These observations motivate several open research directions for applying NNs in IoBNT networks.

1) Alternative NN Architectures: Recent literature has primarily considered feedforward NNs and RNNs architectures for MC systems; see Fig. 10. In contrast, other architectures that have gained significant traction in the broader NN literature—most notably transformers and reinforcement learning—have only recently begun to attract attention in the context of MC. Transformer-based attention mechanisms have been highly successful in domains such as language processing, enabling joint alignment and decoding by capturing long-range dependencies in sequential data [91]. However, despite initial examples discussed in Sec. III-C3, current transformer implementations are not yet tailored to the specific interdependencies induced by ISI in MC channels. As a result, transformer-based architectures require substantial adaptation to fully exploit ISI-driven sequence structure in MC decoding, as also noted in [124].

Furthermore, NNs remain ill-suited to nonlinear scenarios, e.g., turbulent flows, and require vast amounts of data to converge, often with poor accuracy. Nevertheless, given that the underlying *equations* governing the system are known, physics-informed NN models arise as a promising approach. This architecture can include turbulence models by integrating Reynolds-averaged Navier-Stokes or Fokker-Planck equations into the learning algorithm, referring to examples in [125], [126].

Table I – continued from previous page

MC Geometry	NN architecture	Application	End-to-end channel-related parameters: Communication range	Receiver radius	D [$\mathrm{nm^2/ns}$]	Communication-related parameters: Released molecules	Symbol duration	SNR [dB]	Performance metrics		Ref.
Experimental testbeds – Vessel-like channels	RNN	Synchronization	50 cm		0.1	10^3	20 s	30	P_d	0.6	Sec. III-B4
									STO	4 s	
	Transformer	Data communication	< 1 m				1.2 s		BER	$\geq 9 \times 10^{-2}$	[80]
	BiRNN		< 1 m				1.2 s		BER	$\geq 7.4 \times 10^{-2}$	[83]
			< 1 m				250 to 500 ms			$> 10^{-4}$	[81], [82]
	CNN		5 cm				0.25, 0.5, 1 s		Accuracy	$\geq 33\,\%$	[87]
	Feedforward NN						0.5, 1, 2, 3 s		BER	$\geq 4 \times 10^{-1}$	[85]
		Channel estimation	1 to 2 m				250, 500, 750 ms		RMSE	20.1	[39]
Experimental testbeds – Open air	BiRNN	Data communication	1 m		0.84	4.9×10^{23}	1 to 6 s	27.5	BER	$\geq 10^{-6}$	Sec. III-C4
in-vivo bacteria colony	CNN	Data communication	100 µm	6.7 µm	0.75	10^5 bacteria	4 s	10 to 35	BER	$\geq 10^{-4}$	[90]
							1 min			$\geq 10^{-2}$	[89]
Human vessels	Feedforward NN	Channel estimation							RMSE	$\geq 5 \times 10^{-2}$	[123]
		Detection	≈ 2.5 m			10^3			Accuracy	$< 85\,\%$	[109]

Notes:
- In the Ref. column of the table, the entries referring to "Sec." refer to the corresponding section in the present manuscript.
- Missing entries in the table are not specified in the corresponding reference.
- The column $D\,[\mathrm{nm^2/ns}]$ refers to the diffusion coefficient of molecules.
- The column "Released molecules" refers to the number of molecules released at the emitter.
- Few contributions report the channel memory. The work in [78] reports a channel length of 6 symbols. The work in [80] reports 3, 5, 13. The work in [85] reports 1. Our solution in Sec. III.C.4 reports a channel memory of 7 symbols.
- All reported "Data communication" schemes uses the OOK modulation except for [87] which report multilevel modulation as well.
- The acronyms RMSE in the table refers to root mean squared error.
- The entry table for reference in [38] also develops a MC geometry comprised of a volumetric transmitter. Besides, we evaluated the RMSE metric by inspecting the printed Fig. 3 on the same paper.
- The flow velocity corresponding to the entry in the Table "Drifted channel" is $30\,\mu\mathrm{m/s}$ as follows from reference [68], $5.5\,\mathrm{cm/s}$ as in [100], and $10\,\mathrm{cm/s}$ as in the entry for Sec.III.B.4. Additionally, the entry table "Open air" also specifies a flow velocity of $3.5\,\mathrm{m/s}$ for Sec. III.C.4 in the primary document.
- The "RNN" and "BiRNN" entries in the table implement LSTM cells.
- The entry table for the references [66], [73], [74] refer to a mobile scenario where the diffusion coefficient of the emitter is $4.74 \times 10^{-5}\,\mathrm{nm^2/ns}$ and for the receiver is $2.31 \times 10^{-3}\,\mathrm{nm^2/ns}$. Similarly, the entry table for the reference in [71], [75], [76] define a diffusion coefficient of the receiver as $10^{-5}\,\mathrm{nm^2/ns}$.
- The number of released molecules within the entry table of reference [109] refers to the number of nanosensors instead.

2) Hyperparameter Tuning: Within the MC literature, only a few references [66], [68], [77] investigate the minimization of the BER by optimizing the training and architecture-related hyperparameters. The performance of neural network models in MC systems strongly depends on the selection of the training hyperparameters. These include parameters such as the learning rate, batch size, optimizer type, number of training epochs, and regularization strategies, which directly influence convergence behaviour and generalization capability. Looking after the optimization methods, the authors of [68] report using a grid search method to find hyperparameters related to the dimension of the NN's input and the total number of encoders and decoders in the transformer architecture. A similar procedure is illustrated in [66] to find the number of layers and nodes per layer in a feedforward NN. More exhaustively, the authors in [77] use Bayesian optimization to find training parameters, including learning and dropout rates, as well as the number of layers in the CNN and BiRNN architectures. However, the choice of these hyperparameters across the large number of reported papers is empirical and varies significantly with the MC scenario, dataset size, and network architecture. To provide a consolidated overview of configurations adopted in prior studies, Table III summarizes the reported hyperparameter settings used to train neural network models for MC applications. The table highlights the diversity of training strategies and illustrates the lack of standardized configurations across the field. In this regard, several promising research avenues remain open: (i) The extensive body of work on hyperparameter optimization methods has yet to be systematically applied to MC channels (e.g., Bayesian, evolutionary, or gradient-based strategies [127]); (ii) still existing studies mainly focus on the detection task, leaving channel estimation, synchronization, and higher-layer

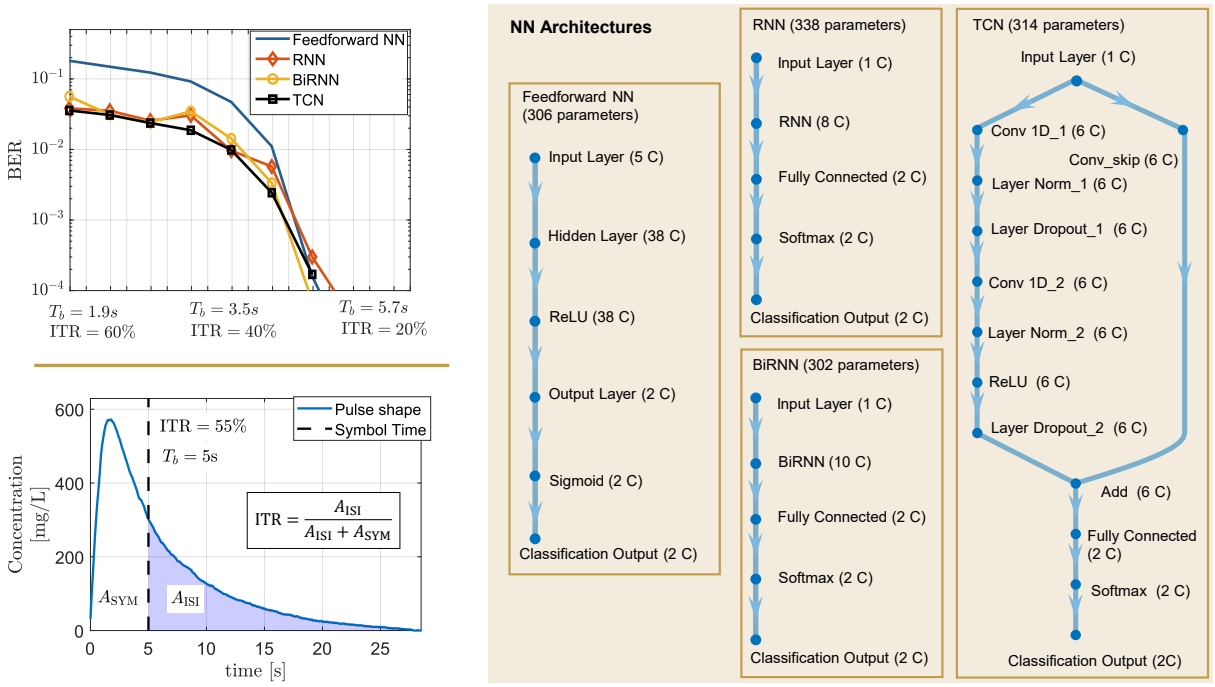


Fig. 11: Graph representation and performance of various NN architectures. 'C' denotes the number of input channels per layer, A_{SYM} and A_{ISI} evaluate the area below the curve.

processes largely unexplored; and (iii) current approaches are mostly agnostic to the underlying MC link, optimizing for BER without accounting for the physical characteristics and dynamics of molecule propagation.

3) Channel Access in MC-based IoBNT Networks: Channel access strategies face the problem that the number of nodes in the link is unknown, which is the case in nanonetwork formation in MC scenarios. For instance, effective disease treatment relies on coordinated nanosensor actions, enabling the controlled release of drugs toward targets such as cancer cells [128]. Protocols have been designed to enable these applications, as seen in [129]; however, the dynamics of MC channels in fluidic environments hinder their straightforward deployment (see [62]). Cognitive strategies developed for node clustering in computer networks, such as those in [130], can be repurposed in MC environments for nanosensor cluster formation as well.

4) Leveraging Cognitive Radio Concepts within the IoBNT Framework: Cognitive radio, originally developed for adaptive wireless systems [131], can be naturally extended to MC environments. As discussed in Sec. III, early research already hints at this integration but remains incomplete. Advancing cross-layer and cooperative strategies could substantially enhance MC performance. At the system level, cognitive units may adapt sensing and transmission to support goal-oriented tasks, such as mimicking bioprocesses in cellular-scale digital twins [132], thereby requiring unified cognitive control across all MC layers.

IV. Enabling Biocomputing in IoBNT Networks

Having reviewed the main methodological and algorithmic approaches, we now shift our focus toward their practical deployment. In particular, this section discusses how the previously described techniques translate into experimental platforms, emerging bio-nanotechnological systems, and real-world applications. This trend motivates the search for biocompatible computing substrates, as future IoBNT nanonodes are expected to support smart processing tasks while remaining compatible with biological environments [9]. To guide this discussion, Fig. 12 provides a conceptual overview of promising substrates for realizing NN components in bio-nano systems, organized according to the main stages of a neuron: molecular or electrical input interfacing, weighted summation, nonlinear activation, and output readout. Bio-inspired micro/nano computer design is a relatively young research field, and the literature reflects varying levels of readiness for individual concepts, which we summarize in this section. Developments in this area lay the foundation to support the intelligent capabilities of the nanocomponents of IoBNT networks.

As summarized in Fig. 12, already investigated technologies are developed along three main directions:

(i) Extending the biological role of DNA-based chemical reactions within the cell for computing; see the evolution in [33, Fig. 1] and [133, Fig. 2a],
(ii) developing chemical reaction networkss (CRNs) [134], and
(iii) engineering microfluidic circuits for computing [135].

Towards the development of NN models, a major trend in the literature is the use of engineered gene regulatory networks (GRNs) (see Sec. IV-B1) due to their programmability for complex calculations [33, Sec. 4.3]. Primarily developed within the communication-engineering community, a second trend involves integrating chemical reactions within microfluidic circuits. Following these research streams, we trace the early developments in digital logic and digital signal processing

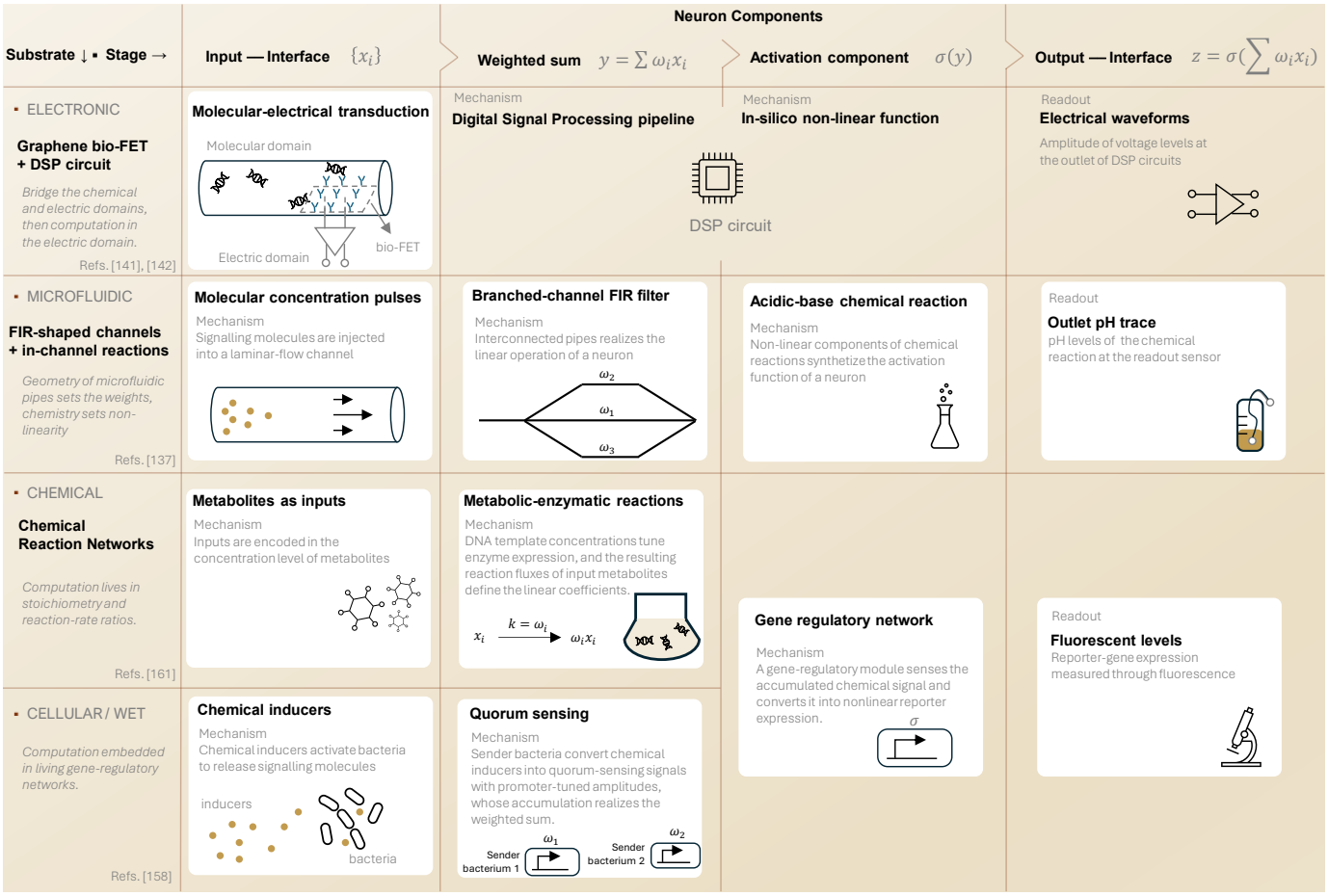


Fig. 12: Summary of biocompatible technologies for NNs deployment.

(DSP) operations in biocompatible substrates for the development of NN modules.

A. Biocomputing in the Digital Domain

In the digital domain, reported research emulates neurons through chemical implementation and integration of logic gates. Basically, a set of excitatory and inhibitory chemical reactions is connected by common substrates to realize logical operations such as `AND`, `OR`, and `NOT`.[12] The first description of CRNs, realizing neuron components of McCulloch–Pitts neurons,[13] dates back to the work in [136], where a neuron was prototyped through an enzyme reaction system in free-diffusion environments. Subsequent work develops neurons by connecting logic gates, in which a 2-neuron feedforward NN is realized in [134] to classify black-and-white images. The linear neuron operation results from mixing encoded bit-1s and bit-0s, analogous to the mixing of acid and base compounds. The neuron's non-linearity is implemented via pH-based thresholding. The classifier identifies 8×8 resolution images of handwritten digits with an accuracy larger than $95\,\%$.

The literature also reports models for DSP operations in microfluidic channels, where the analytic formulation is valid under laminar flow conditions [137]. A finite impulse response (FIR) filter is modeled in [137, Sec. V] to design lowpass, stopband, and bandpass filtering of molecule concentration levels.[14] The impulse response (in the time domain) and transfer function (in the frequency domain) are derived for combined microfluidic straight channels of arbitrary length, turns, bifurcations, and junctions. Although it was not originally conceived for implementing neurons, the FIR-based microfluidic circuit design provides a basis for realizing neuron-like operations in NN architectures. For instance, a 1D CNN operation, $\sum_i \omega_i x_{(i-n)}$, maps naturally onto an FIR structure by treating the filter coefficients ω_i as learnable parameters. The neuron's nonlinearity, corresponding to the activation function, may then be implemented via chemical reactions embedded in the microfluidic channel, thereby completing the weighted-sum/activation pipeline illustrated in the microfluidic row of Fig. 12; related nonlinear chemical mechanisms are reported in [138], [139], [140].

Complementing purely microfluidic realizations, the electrical domain provides a mature substrate for implementing the DSP components of NNs, such as adders, multipliers, and nonlinear functions. In this approach, molecular concentration patterns are first transduced into electrical waveforms, allowing subsequent weighted-sum and nonlinear operations to be exe-

[12] A CRN is an abstraction of the dynamics of a system of chemical reactions given by a finite set of differential equations; see [33, Sec. 3.1].

[13] McCulloch–Pitts neurons have only two possible states, either firing or quiescent.

[14] Calculations are reported in [137] to be accurate whenever the flow is laminar, i.e., at small Reynolds numbers, see [23, Eq. (13)].

TABLE II: Complexity of reported NN architectures.

MC Geometry		NN architecture	Application	Input length	Fully connected layers			RNN layers		CNN layers			Number of parameters	Ref.
					FC layers	Hidden layers	Nodes per hidden layer	Number of layers	LSTM cells per layers	Number of layers	Number of filters	Filter size		
Point transmitter-Free diffusion-Spherical absorbing receiver		Feedforward NN	Channel estimation	2	1	1	2						6	Sec. III-A4
				3	1	5	10, 20, 40, 20, 10						2163	[43]
			Localization	10^3	1	5	400						1 041 603	[115]
			Detection	16	1	1	16						272	[120]
			Data communication	6	5	10	4						837	[64]
				11	1	2	20						722	[73], [74]
				11	1	12	10						1242	[66]
				120	1	2	70 and 10						9111	[67]
				12	8	1	200, 5, 1			3	3	10, 3, 1	15 688	[76]
		RNN	Detection	32	1	1	32	1	32				417	[121]
Experimental testbeds	Open air	BiRNN	Data communication	8				6	8				576	Sec. III-C4
	Vessel-like channels			5	2	1 and 1	25	2	5				540	[83]
				40				32	40				2880	[82]
		RNN	Synchronization	1	2	1 and 1	256 and 256	1	128				35 328	Sec. III-B4
		CNN	Data communication	128	3	1	4096, 4096, 6			3	64, 2, 2	7, 5, 3	33 587 738	[87]
in-vivo bacteria colony				$15 \times 60 \times 2$	2	1	32 and 2			1	16	15	462 144	[89]
				5						2	2	3	30	[90]
Human vessels		Feedforward NN	Detection	2	1	1	6						18	[109]
Drifted channels		Autoencoder	Data communication	1	Enc: 3 Dec: 5	Enc: 1 Dec: 1							244	[100]

Notes:
• The calculation of the number of elements per layer is evaluated as follows:
- Feedforward NN: We compute the total of parameters by adding the number of coefficients and bias per layer. Per layer, the total of coefficients is $n_{\text{in}} \times n_{\text{out}}$, and the total of biases is n_{out}, where n_{in} is the number of inputs, and n_{out} is the number of outputs for the given layer. The number of nodes in the output layer is 1, except for the entry in [66], where the nodes in the output layer are 2.
- RNN and BiRNN architectures: We compute the number of parameters based on the gates within the LSTM cells, as all reported methods deploy them. The LSTM cell comprises four gates: input, forget, cell, and output. Each gate implements a separate set of weights for the input, hidden states, and biases, with eight weights and four biases, totaling 12 parameters per cell. In total, the amount of parameters for n_{cells} cells will be $n_{\text{cells}} \times (12)$. In the case of the BiRNN, calculations are twice the number as for the RNN, as each layer implements both a backward and forward direction.
- CNN architecture: The filter size gives the number of parameters. A filter of size K will implement K weights and a single bias. Additionally, the total number of filters is determined by the comparative sizes of the input and output. For instance, if the input is a 1D vector as 128×1 and the output is a 2D matrix as 128×64, the number of filters is 64. If the input and output are 2D matrices, given by 64×64 and 64×128, respectively, then two filters would be needed, as the output dimension is twice the input dimension.
- Transformer architecture: The complexity for this architecture is evaluated based on the code published by the authors in [76]. In this entry, the number of nodes per hidden layer, 200, refers to the six fully connected layers implemented in the transformer architecture, while the other values, 5 and 1, refer to the remaining fully connected layers implemented in the decoder. This architecture also deploys three CNNs at the Transformer input to extract features.

TABLE III: Summary of hyperparameters related to the training of NN architectures.

MC Geometry		NN architecture	Application	Optimizer algorithm	Learning rate	Number of samples training	Number of epochs	Batch size	Ref.
Point transmitter-Free diffusion-Spherical absorbing receiver		Feedforward NN	Channel estimation	BFGS		481 samples	15		Sec. III-A4
			Localization	Adam	10^{-2}		100	256	[115]
			Detection		10^{-3}	10^5 samples	100	10	[120]
			Data communication	LM	10^{-2}	10^3 bit	200	50	[78]
						5×10^4 bit		1×10^3	[64]
				BFGS	10^{-3}	5×10^4 bit	50, 55, 125		[73], [74]
							184, 188, 215		
						10^4 bit			[66]
					10^{-3}		500		[67]
		RNN	Detection	Gradient descent	10^{-3}	8×10^3 samples	100	10	[121]
Experimental testbeds	Open air	BiRNN	Data communication	Adam	10^{-3}	10^6 samples	10	10	Sec. III-C4
	Vessel-like channels					8×10^4 bit	2×10^4		[83]
					10^{-3}	120 bit	200	10	[82]
		RNN	Synchronization	BFGS	2×10^{-4}	1.2×10^6 bit	35×10^3		Sec. III-B4
		CNN	Data communication	Adam	10^{-3}		20	64	[87]
in vivo bacteria colony		CNN		SGDM					[89]
Human vessels		Feedforward NN	Detection				4×10^4		[109]
Drifted channels		Autoencoder	Data communication		9×10^{-3}		2×10^3	40	[100]

Notes:
• In the Ref. column of the table, the entries referring to "Sec." are pointing to the corresponding section in the main document.
• The abbreviations LM, BFGS, and SGDM introduced in the column "Optimizer algorithm" refer to Lavenberg-Marquardt, Broyden-Fletcher-Goldfarb-Shanno (see [141]), and stochastic gradient descent with momentum.
• Missing entries in the table are not reported in the corresponding manuscript.

cuted by conventional electronic circuits. Graphene biofield-effect transistors (bio-FET) can provide this molecular–electrical interface by detecting DNA strands flowing in microfluidic channels and converting them into electrical signals [142], [143]. This pipeline is illustrated in the first row of Fig. 12; although experimental demonstrations in IoBNT networks remain unexplored, bio-FET sensing technologies represent one of the more mature routes for molecular–electrical interfacing.

B. Biocomputing in the Analog Domain

Unlike *in silico* technologies, where NNs are deployed digitally, the integration of NNs with chemical-based technologies is more naturally pursued in the analog domain. This is mainly because chemically wiring pre-existing logic gates in fluidic media to implement more complex arithmetics remains challenging. The following paragraphs summarize several reported NN implementations that are based on compartment-based models, GRN, and biological *in vivo* systems.

1) Wet Neuromorphic Computing

Computing can be realized in the analog domain using the already-in-place neuromorphic capabilities of biological systems [144], [145].[15] For instance, within the cell, memory and computation are inherently integrated into the GRN (which refers to neuromorphic computing), enabling regression and pattern recognition tasks, as illustrated in [146]. The concept also extends to offloading computation into tissues and entire organisms; see the most vivid examples in [147], [148].

Within the cell, arithmetic can be realized in the analog domain by engineering DNA circuits, as summarized in [33], [149], [150]. Examples include adders, multipliers, subtractors, power-laws, and dividers realized by combining DNA circuits, as illustrated in the early work in [151]. To a greater extent, the literature focuses on implementing feedforward NN architectures that leverage the natural interactions among genes within the cell. For instance, NNs are identified with genes as the computing units, transcription factors as the stimulus, and the degree of influence of transcription factors in the genes as the weights [152], [153], [154]. Using these analogies between DNA-chemical components and NN functions, the potential of operating NN structures already within the gene regulatory circuits is extensive –as a result of mining the connection in the GRN, more than a hundred single-layer NNs have been identified, see [150, Fig. 3].

The first work demonstrating the potential of DNA-based engineering relied on the DNA strand-displacement technique as developed in [155].[16] In this technique, neurons are implemented via the reaction of input strands and toeholds (DNA segments) with three gates. The multiplier gates output a number of DNA strands in proportion to the weight, i.e., the operation $\omega_i \times x_i$, where ω_i is the weight and x_i is the input. The integration gate is chemically related to the output strands from the multiplier gates in a way that a reaction product reflects a strand as the sum of the inputs, i.e., $\sum \omega_i \times x_i$. Finally, thresholding is implemented by a dedicated threshold gate that releases output strands only when the accumulated signal exceeds a prescribed level; these output strands are then detected through fluorescence from DNA reporter gates. Using 112 DNA strands assembled into 72 initial DNA species, this technology implemented a four-neuron Hopfield associative memory and demonstrated pattern recall through a "read your mind" game with four stored patterns.

On a larger scale, other examples include combining multiple engineered cells [157], [158] and the striking performance of brain organoids cultured cells [159]. The research in [157] conceives a single-layer NN via cell-to-cell communication in a diffusive medium. The weights of the NN are set by the diffusive properties of the information molecules. The required nonlinearity is achieved through gene expression in an engineered *E. Coli* cell. Integrating 2 to 6 bacterial populations, each population acting as a 2-input neuron, this work demonstrated the implementation of multiplexers, majority rules (data fusion), and decoding, among other operations. A different approach is developed in [158], in which multiple bacteria serve as transmitters and release molecules proportional to the magnitudes of the learnable coefficients of the NN. Another bacterium serves as a receiver and implements the nonlinear function; see an illustration of the complete pipeline in the bottom row in Fig. 12. The concept's applicability to ML was demonstrated in [158] through a pattern-recognition task, in which an engineered bacterial population, performing as a neuron of 9 learnable coefficients, classified chemical-inducer input patterns arranged as 3×3-bit images. In both implementations [157], [158], the biochemical computation is transduced into fluorescent reporter expression, which serves as the measurable readout of the NN output.

Brain organoids, which develop in cultures of human stem cells, are known to perform complex tasks in real-time. The work in [148] illustrates training living neuron cells to play the arcade game 'Pong'. The learning capability of these networks relies on neuronal neuroplasticity. The experimental work illustrates the response of surrounding cells to an electric stimulus and the development of new neural connections during the learning phase. This technology offers two key advantages related to scalability: (i) reduced energy consumption, with the human brain serving as a reference for highly connected yet energy-efficient computation at approximately 20 W, and (ii) faster training; it takes about 4 epochs to match the performance of LSTM cells, which require 50 epochs.

2) Analog Computation with CRNs

DNA regulatory circuits are also in the domain of CRNs, as they provide the substrate for the reaction pathways.[17] However, other chemical solutions implement the computation without engineering DNA strands. These solutions aim to overcome limitations such as temperature sensitivity, long computation times, and the use of a large number of chemical components, as stated in [160]. For instance, the metabolic circuits of cells are engineered in [161] to implement 2 and 4–input

[15]Neuromorphic computing denotes computing principles inspired by biological nervous systems, in which information processing emerges from distributed interactions among interconnected units and memory is embedded in the system dynamics, rather than being separated from computation as in conventional von Neumann architectures.

[16]DNA strand displacement regulates the gene expression and is capable of creating universal computation [156].

[17]Substrates refer to reactant compounds in a chemical reaction.

NNs of a single layer. This work achieved positive weights with concentration level in a cell-free framework and negative weights with attenuating reactions. However, due to the large number of required reactions, these techniques are often combined with DNA-computing [162], [163], [164] to facilitate implementation. Notably, CRNs were used to implement a chemical NN that even allows backpropagation and, thus, online in-situ NN training [165].

3) Compartment-Based Models

Nanoscale computing units can be realized through the propagation of molecules between connected compartments and the chemical reactions occurring within them [166]. Matrix weights are mapped by adjusting compartment volumes, forming the basis of a compartment-based reaction–diffusion computer, which has been experimentally demonstrated using phase transition, precipitation, and acid–base reactions [167]. These experiments achieved reasonable accuracy, indicating feasibility for microscale implementations. Further theoretical work extends this concept to molecular nano-NNs and has been validated through classification and regression tasks [168].

4) Reservoir Computing in MC Channels

Reservoir computing has recently been introduced in the MC domain to realize recurrent architectures [169], also via brain organoids [147]. This architecture employs a static RNN in a reservoir, with learning occurring only in the output layer [170]. The reservoir is implemented physically as a point-transmitter free-diffusion MC channel with ligand–receptor interactions, which represents the state update equation of the RNN [169, Eq. (1)], or using brain organoids as described in [147]. The output layer is interfaced with a silicon substrate and trained as a weighted vector of bound-molecule samples over time (see [169, Eq. (9)]).

C. Resource Demand and Feasibility of Bio-AI Unit Implementation

The required arithmetic operations for NN operation, such as adders, multipliers, and non-linear functions, are highly resource-demanding to be implemented by logic units within cells or microfluidic devices. For instance, an eight-bit adder would require 16 `AND`, 8 `OR` and 8 `XOR` two-input gates; see [171, Sec. 5.2.1]. The reported NN-based decoder architectures in Sec. III require around 300 learnable parameters (see Fig. 11) and nearly the same number of adders, yielding a total of $300 \times (16 + 8 + 8) = 9600$ logic gates to be implemented. Enforcing a gate per cell or per microfluidic circuit imposes a physical dimension on the millimeter scale, preventing low-dimensional developments at the micrometer or nanometer scale.[18] Furthermore, hundreds of learnable parameters lead to a high number of connections, imposing a greater challenge for the chemical wiring of the corresponding logic units.

By contrast, GRNs within the cell allows a higher computational density. Following the entries in [150, Fig. 3] related to feedforward NN implementation, we can find networks of 26 learnable parameters maximum.[19] Therefore, 14 connected cells can achieve the complexity reported for decoders (approximately 300 in Fig. 11) and still fit within the micrometer range. However, the limited reliability of DNA circuits [33] and their long processing times (on the order of hours [163, Fig. 4]) remain significant drawbacks. Beyond purely genetic circuits, brain organoids [159] and reservoir computing [169] are candidate technologies for developing NN architectures at the micrometer scale. Reported designs are in the range of 10 to $100\,\mu\mathrm{m}$, although not limited to further scaling down their dimensions. These advancements demonstrate the practicality of NN architectures for IoBNT applications; yet, a design for NN module interfaces (both biological and electrical) remains unavailable.

A comparative summary of the technologies presented in this section is provided in Table IV, which evaluates their capabilities in terms of weight density, latency, scalability, and technology readiness level (TRL).[20] To ensure a fair comparison across diverse substrates, we evaluate computational capabilities using spatiotemporal weight density (number of weights per unit volume per unit time). This metric is necessary because a purely spatial evaluation tends to overestimate the efficiency of silicon technologies. That is, in contrast to chemical-based technologies, silicon enables iterative hardware reuse through code loops. By accounting for the reusability capacity of *in silico* technologies, this evaluation provides a realistic sense of the physical scale required to synthesize NN models. The values are extracted from the experimental studies cited in the last column of the table, with specific evaluation details for weight counts and volume, as provided in the table notes.

As a result of these calculations, the highest TRL is achieved with bioFET technology, see also a summary of similar biosensors on the market in [175]. Chemical technologies correspond to lower TRL levels, where prototypes have been developed, and some have been tested in relevant scenarios. We find that among the chemical-based technologies, compartment-based technologies have the highest gain, although it is six orders of magnitude lower than that of silico technologies. We also remark that the lower densities of the chemical technologies are primarily due to the reported experimental volumes being on the order of tens of µL, which understate their weight-density capacities. These contributions illustrated the affordability of developing NNs with bio-compounds, but less attention was paid to their efficiency in terms of size and performance. This motivates the development of more size-efficient proofs of concept.

D. Concluding Remarks and Future Outlook

In summary, there are numerous approaches to implementing NNs in biological and/or chemical contexts. A key challenge lies in the limited scalability of these concepts, constrained by geometrical [167], chemical [165], or biological [176] constraints. While numerous concepts have been shown to work in practice, a unified framework for the design and standardization of biological or chemical NNs is currently missing. The development of such a unified scheme is anticipated, as current research in the field is advancing.

[18] We assume the dimension of a cell is in the range of 1 to $2\,\mu\mathrm{m}$ as for *E. Coli* [172] and for microfluidic circuits in the range of 1 to $10\,\mu\mathrm{m}$ [173].

[19] We perform this calculation assuming the number of hidden nodes follows the number of inputs.

[20] For a definition of readiness levels, see [174, pag. 14].

Besides, we identify the following open research directions related to the implementation of NNs in the biological domain:

1. MC Networks as ML Platforms: Inspired by paradigms in WiFi and ML joint development [56], MC networks can also constitute a running platform for ML development (see [99, Chap. 12]). Various examples exist in wireless networks where intelligence is distributed among nodes. Such ideas can be extended to MC networks for deploying large NN architectures challenging to fit on a single node.

2. Deployment at the Nanoscale Level: The current integration of MC and ML primarily relies on macro-scale external computers hosting the NN inference (Sec. III). The micro-scale implementations illustrated in this section still remain limited in complexity. A promising direction for reducing resource consumption is to develop logic based on a one-bit digital representation of molecular signals, such as delta-sigma modulation schemes [177, Sec. 12.3]. This approach could lead to digital-like biocomputers based on molecular analog-to-digital converters and low-bit-precision logic elements—such as adders and multipliers—implemented via chemical or biological reactions [178], [179], [180], [181], [182].

V. Early Work on Explainable Neural Networks

Over the past few decades, MC has evolved from theoretical concepts to practical implementations, leveraging advances in nanotechnology and bioengineering [29], [30]. Concurrently, the integration of AI has transformed MC by enhancing the efficiency and robustness of signal processing, encoding, transmission, and decoding processes, as discussed in [31]. Despite these advancements, the complexity and opacity of AI models pose a significant barrier to their full realization in MC systems, as proof of correctness is lacking. The directions presented in this section address these challenges by introducing methods to make the application of NNs transparent. This section outlines the motivation behind this research direction, followed by a brief description of the fundamental tools, a summary of applied XAI in the context of IoBNT networks, and an illustrative code example.

A. Motivation

The application of NNs in MC has traditionally focused on optimizing various aspects of the communication process, see examples in Sec. III. However, the "black-box" nature of these models often leads to a lack of transparency and interpretability. Hence, the traditional application of NNs can be limiting in critical applications where understanding the decision-making process is crucial. This is where explainable artificial intelligence (XAI) comes into play, offering a suite of techniques designed to elucidate the inner workings of AI models, making them more transparent and comprehensible.

The motivation behind XAI research in MC is twofold [31]. Firstly, there is a pressing need to bridge the gap between the sophisticated NN algorithms used in MC and the requirement for transparency and explainability, particularly in healthcare applications. XAI can help provide evidence supporting the correctness of deployed NN-based models, thereby increasing confidence in their operation. Moreover, XAI can facilitate interdisciplinary collaboration by making NN models more accessible to researchers and practitioners across biology, chemistry, and engineering. Interpreting the operation of NN modules is ultimately a translation of abstract neural weights into physically meaningful insights rooted in biology and chemistry, a process *per se* that requires multidisciplinary expertise.

B. Explainable and Interpretable Molecular Communication

The interpretability of a model is limited by a balance between its complexity and transparency requirements [187]. Models with fewer parameters and near-linearity become more interpretable, albeit with limited performance. Performance and explainability are counteracting; balancing these two aspects poses significant challenges in designing high-performance methods for sensitive use cases. Within the scope of this challenge, a variety of metrics are used to evaluate the effectiveness of explanations, as described next.

1) Explainability Metrics

The following metrics evaluate to what degree NN-driven predictions can be trusted and understood within the intricate dynamics of nanoscale interactions [188]:

Fidelity: Fidelity quantifies how accurately an explanation with a surrogate model reflects the behavior of the original model. A surrogate model serves to explain the primary model's predictions, and fidelity measures the alignment between them. A high-fidelity explanation implies that the surrogate model accurately captures the core decision-making processes of the original one. This is particularly crucial in MC, where understanding the molecular mechanisms that drive predictions can shed light on the interactions at the nanoscale level.

Sparsity: Sparsity reflects the simplicity of an explanation by quantifying the number of features involved. Higher sparsity corresponds to fewer features being included, leading to more concise and interpretable explanations. In the realm of MC, where interactions at the nanoscale level are inherently complex and multifaceted, sparse explanations are particularly valuable to identify the most relevant physical processes that influence predictions.

Stability: Stability measures the consistency of an explanation when the input is subject to small perturbations. In MC, where environmental factors and molecular interactions are dynamic, stable explanations ensure that the insights provided by the NN model are reliable and robust across different scenarios.

Causality: Causality assesses whether the features identified in the explanation are genuinely responsible for the model's output. In the context of MC, this involves identifying whether specific signaling molecules or interactions directly influence the predicted cellular responses. This metric is particularly valuable for validating experimental designs and ensuring the reliability of insights in complex molecular networks.

Comprehensibility: Comprehensibility reflects how easily a human can interpret the explanation. Although difficult to quantify directly, it can be associated with the simplicity of the explanation. For example, a shorter rule-based explanation with fewer conditions is generally easier to understand than a complex one.

TABLE IV: Comparison of reported bio-computing technologies.

	Technology		Domain	Weight Density $[\text{weights}/(\text{mm}^3\,\text{s})]$	Latency	Scalability (approx.)	Bio-compatibility	TRL	Ref.
Digital only	Graphene bio-FET		Silico	10^5	milli seconds	High	Medium	TRL 6-7	[142], [143]
Digital & Analog	Chemical reaction networks		Chemical	10^{-4}	minutes	Low-Medium	Medium–high	TRL 3-5	[161] [163], [160]
Analog only	Compartment-based		Chemical	10^{-1}	seconds	Medium-High	Medium–high	TRL 4-5	[166], [168]
	Wet Neuromorphic Systems	DNA Strand Displacement	Chemical	10^{-4}	hours	Medium	Very high	TRL 3-4	[155]
		Bacteria Colony	Cellular	10^{-6}	hours	Medium	Very high	TRL 2-3	[157], [158]

Notes:
• The "Computational Density" column denotes the areal concentration of NN weights that can be physically hosted per unit volume and evaluated per unit of time. To evaluate the computational density per entry, we followed the remarks below

- **Graphene bio-FET:** In this case, we assume that a digital back-end circuit is attached to the graphene bio-FET, which serves as the interface between the molecular and electrical domains. This digital circuit incorporates an analog-to-digital converter (ADC) for signal conversion and a microcontroller unit that hosts the on-device NN model. Following TinyML solutions targeting healthcare applications, as reported in [183], we selected the work in [184] as an illustrative high-complexity example. This solution deploys a CNN of 197 596 parameters within the microcontroller nRF52832 of dimension $3\,\text{mm} \times 3.2\,\text{mm} \times 0.225\,\text{mm}$[185]. The reported processing time is $97.8\,\text{ms}$ [184, Sec.IV.C], resulting in a parameter density of $\frac{197596}{3\,\text{mm}\times 3.2\,\text{mm}\times 0.225\,\text{mm}\times 97.8\,\text{ms}} \approx 9.2\,\text{K}\,\text{weights}/(\text{mm}^3\,\text{s})$.
- **Chemical reaction networks:** The reference in [160] reports three fully connected neurons to process input images of size 8×8, 12×12, 16×16, and 28×28, yielding a total of weights per image dimension as 64, 144, 256, 784 ([160, Tab. 2-41, Supplement]). The reaction volume is reported as $2\times 2.4\,\mu\text{L}$ per weight, yielding a total volume per image dimension as $307.2,\ 691.2,\ 1228.8, 3763.2\,\mu\text{L}$. The reaction times per image dimension as $14,\ 32,\ 57, 174\,\text{min}$. Following these values, the density per image dimension yields $2.5\times 10^{-4},\ 1.1\times 10^{-4},\ 6.1\times 10^{-4}, 2\times 10^{-5}\,\text{weights}/(\text{mm}^3\,\text{s})$.
- **Compartment-based:** The contribution in [166], refers to a total of 6 parameters (3×2 matrix multiplication in [166, Eq. (71)]) synthesized in a total volume of $25\,\mu\text{m}^3$ (see [166, Tab. II]). The latency to develop the matrix multiplication yields approximately $3\times 0.4\,\text{s} = 1.2\,\text{s}$, see the [166, Fig. 14]. This figures yield a density of $\frac{4}{20\,\text{mm}^3 1.2\,\text{s}} \approx 0.16\,\text{weights}/(\text{mm}^3\,\text{s})$. Similar order of magnitude holds for the work in [168], where the number of parameters is 8 [168, Eq. (23)], the reaction volume is $15\,\mu\text{L}$ ([168, Tab. II]) and latency is approximately $2.5\,\text{s}$ ([168, Fig. 3]), yielding a density of $0.2\,\text{weights}/(\text{mm}^3\,\text{s})$.
- **DNA Strand Displacement:** We base our calculations on the experimental design of the four-neuron Hopfield associative network introduced in [155, Fig. 3]. A total of 12 weights are synthesized with DNA strands [155, Supp. S4] in a reaction volume of $1\,\text{mm}^3$. The circuit operates with a latency of $8\,\text{h}$, see [155, Fig. 3 d]. These figures yield a density of $4.2\times 10^{-4}\,\text{weights}/(\text{mm}^3\,\text{s})$.
- **Bacteria Colony:** The contribution in [157] develops a bacteria colony, where 9 weights are synthetized using 5 bacteria, see [157, Fig. 5c]. The processing time takes up to $16\,\text{h}$, see [157, Supp. So]. Experiments are reported on 96-well microplate, which we assume with the minimum work volume of$25\,\mu\text{L}$, see [186]. Following these figures, the calculations for the density yields $\frac{9}{5\times 25\,\mu\text{m}^3\times 16\,\text{h}} \approx 1.3\times 10^{-6}\,\text{K}\,\text{weights}/(\text{mm}^3\,\text{s})$. A similar order of magnitude holds for the report in [158], where 8 weights are synthesized in a 96-well microplate and the calculation span in the range 2 to $5\,\text{h}$.
- **Reservoir Computing:** These contributions develop a reservoir computing architecture. Following the contribution in [169], the learning phase is given by the vector $\mathbf{W}_{\text{out}}$ in [169, Eq. 2], which boils down to performing a weighted sum. As such, the density for this approach relies on the means to develop a vector multiplication through any of the listed entries in this table. Similar calculations account for the contribution in [147]. As such, we avoid listing Reservoir Computing as a separate row in the table.
- **Brain Organoids:** We avoid listing the calculations for the contribution in [148] as there isn't an explicit association of learnable coefficients to the connection among cells.
- **GRN:** We are not listing GRN as we couldn't find literature that follows an experimental design.

2) Explainability Techniques

Benefiting from the above-mentioned metrics, there are several key techniques for providing explanations in NN models, as summarized below:

Feature Attribution Methods: These methods assign an importance score to each feature. The most common techniques include local interpretable model-agnostic explanation (LIME), shapley additive explanation (SHAP), layer-wise relevance propagation (LRP), and deep learning important features (DeepLIFT) [187]. LIME operates by perturbing the input data and observing the changes in prediction to create a local, interpretable model. A weighted linear model is then trained with the perturbed data, aiming to approximate the original model; see details in [189].

SHAP values are derived from cooperative game theory, providing a unique solution that allocates payoffs to players in the fairest manner (features). The SHAP value for a given feature is evaluated as in [190, Eq. (4)]. This evaluation considers all possible combinations of input features and computes the marginal contribution of each feature to the model's prediction. In essence, SHAP estimates how much a particular feature, when added to different subsets of features, changes the prediction, and then averages these effects to yield a global contribution score. This mechanism allows for a consistent and locally accurate explanation of feature influence. We develop an illustrative example for this method in Sec. V-D.

LRP is a method designed to attribute an NN's prediction back to its input features [191]. This is achieved by backpropagating relevance from the output layer through the network to the input features, with relevance determined by distributing contributions in proportion to the weighted connections between neurons. In the context of MC, LRP helps to identify which nanoscale signals or features play the most critical roles in the prediction, enabling better interpretability of complex NN models.

DeepLIFT extends LRP by comparing the actual input to a reference input or baseline, quantifying the difference in model outputs; see further details in [192]. By explicitly accounting for deviations from a baseline, DeepLIFT is particularly useful in scenarios where the relative importance of nanoscale molecular changes can provide crucial insights into the system's dynamics. For instance, in MC systems where slight variations in vesicle concentration, timing, or signal shape may signify different biological states or disease progressions, DeepLIFT can highlight which input perturbations most significantly influence the model's prediction. This enables researchers to trace model decisions back to specific molecular behaviors, such as an unexpected peak in a molecule's release rate or a delay in arrival time, thereby improving interpretability in AI-driven diagnostic or monitoring systems. Unlike gradient-based methods that may suffer from vanishing signals, DeepLIFT preserves contribution scores even in saturated or nonlinear regions, which are common in biochemical signaling cascades.

Saliency Maps: Saliency maps, commonly used in computer vision, highlight regions of an input image that are most important for a model's prediction [193]. This can be helpful in MC, as it visually highlights the most influential features in the NN model's decision-making process. This technique generates visual representations that show which areas of input data contribute most to the model's output, allowing researchers to quickly identify key molecular factors or patterns driving the communication process.

Counterfactual Explanations: These methods provide insights by answering "what-if" questions [194]. For a given input, a counterfactual explanation identifies the smallest input change such that the model output shifts to a target value. Where molecular interactions dictate system responses, counterfactual explanations can help researchers understand the causal relationships between molecular signals and communication outcomes. For instance, counterfactual explanations can reveal how these changes would affect the model's prediction of system behavior by modifying the concentration of a particular molecule or altering a specific signaling pathway. This technique enhances the interpretability of AI models by providing actionable insights into the conditions under which MC might behave differently, helping researchers better understand and control nanoscale interactions.

Manual Permutation Importance: The manual permutation importance (MPI) method is a model-agnostic technique used to quantify the contribution of individual features to a ML model's predictive accuracy. It measures feature importance by randomly shuffling a specific feature's values and observing the resulting increase in model error, thereby effectively disrupting its relationship with the target variable. This approach was first introduced in the context of Random Forests [195], where it was used to assess variable significance by measuring the impact of permuting a variable on prediction accuracy. For MC, where factors such as molecular diffusion, receptor binding, and molecular degradation introduce stochastic variability, explainability methods such as MPI ensure that AI-driven models remain not only accurate but also biologically meaningful. By identifying which features most significantly affect model performance, MPI aids in understanding the underlying biological processes and enhances the interpretability of complex MC models. We develop an illustrative example for this method in Sec. V-D.

C. Reported Research

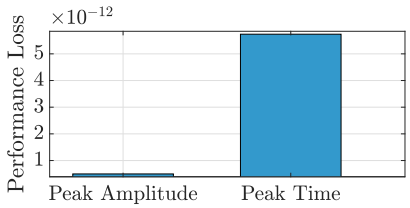
Traditional communication schemes in MC often struggle to evaluate complex end-to-end channel models and minimize the BER. Although reported NN models are a promising alternative to conventional detection methods (see those reported in Sec. III), a critical limitation is their inherent lack of transparency [31]. As summarized below, research studies have begun to address these limitations, with a primary focus on feature-attribution methods to improve interpretability. Existing approaches largely focus on PHY-layer symbol detection. Published research work aims to provide a proof of correctness for NN-based approaches by mapping them to conventional detectors, such as threshold, slope, or linear decoders, see e.g. [104], [196], [197], [198].

XAI methods have been applied to elucidate the inner workings of NN-based symbol detectors for a 2×2 MIMO MC channel in [196]. To interpret the NN's behavior, the authors employed XAI techniques such as LIME, partial dependence plot (PDP), and the individual conditional expectation (ICE) method. Their application revealed that the trained NN effectively operates as a threshold detector and slope detector in the low and high ISI regimes, respectively. This association of the NN with a threshold or slope detector is the model's interpretation and also provides proof of the NN-based detector's correctness.

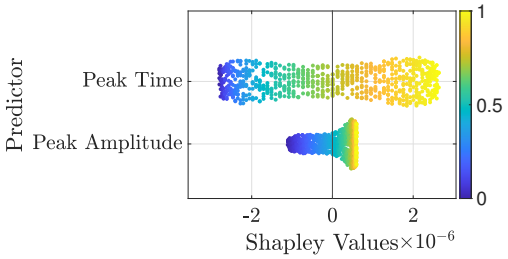
In a similar direction, researchers generated synthetic data from real testbed measurements to train an NN for binary symbol detection in [197]. The key objective was to demystify the "black box" nature of NNs and provide assurance of their correctness in symbol detection tasks. To achieve this, they employed XAI techniques, such as LIME and ICE plots. The analysis revealed that the trained NN's decision-making process closely mirrored that of standard peak and slope detectors in low and high ISI regimes, respectively. The NN behaved similarly to traditional methods used in MC for symbol detection based on signal features such as amplitude peaks and signal slopes.

Another notable development in MC is the integration of the IoBNT framework, where powerful external transmitters communicate with computationally constrained internal receivers. To address this asymmetry, the authors of [104] introduced an AAEC architecture for end-to-end learning in MC systems and analyzed the transmitter using a surrogate model, showing that it behaves similarly to a zero-forcing precoder in low and moderate ISI regimes. This correspondence enables anticipation of the NN's behavior using well-known linear precoding principles while improving the transparency of the learned transmitter.

By integrating SHAP into MC detector design, the authors in [198] identified crucial feature points in the received molecular signal. SHAP-driven analysis highlights the most informative segments of molecular signal waveforms, providing a systematic explanation of how different NN models arrive at detection decisions. Previous work in MC has demonstrated the potential of combining model-based methods with ML to compensate for unknown channel parameters; however, the inherent opacity of deep NNs remains a persistent barrier to



(a) Permutation importance.



(b) Shapley summary.

Fig. 13: Feature importance analysis using manual permutation importance.

practical deployment. This approach maintains the adaptability and high accuracy of data-driven detectors while enhancing explainability, ultimately facilitating safer and more reliable MC applications in areas such as targeted drug delivery and *in vivo* biochemical monitoring.

D. Illustrative Code Example and Results

As a code example, we use the model already developed in Sec. III-A4, which employs a 2-node NN to estimate the distance between immune and cancer cells. The model first extracts two features from the number of vesicles in immune cells: the amplitude and the time coordinate (peak time) of the minimum slope in this sequence; see the block diagram in Fig. 4. The NN inputs are these two features, and the output is the predicted distance, as illustrated in Fig. 4c. Following our previous work in [199], we implemented the MPI method to evaluate the significance of each feature for the NN's output, and also included the SHAP value for further illustration.

The results of applying the MPI method are illustrated in Fig. 13a. The plot highlights the dominant role of the peak position, resulting in a substantially larger performance loss than the peak-amplitude feature. This indicates a stronger correlation with distance and suggests that the peak time encodes key characteristics of vesicle-mediated communication. Complementing this, the SHAP analysis in Fig. 13b shows that peak time spans a broader range of SHAP values—reflecting both stronger and more variable influence on predictions—whereas peak amplitude remains tightly centered around zero. This behavior aligns with the physics of free-diffusion channels, where the travel time encoded in the peak position follows the relation $t_{\text{peak}} \propto d^2/D$ [200, below Eq. (2.8)], where d is the distance between cells and D is the diffusion coefficient of the released vesicles. Overall, the NN architecture effectively focuses on peak timing to capture this fundamental relationship between peak time and distance.

E. Concluding Remarks and Outlook

The transparent operation of technologies is crucial in sensitive applications such as healthcare and medicine, where understanding the inner workings of deployed NNs may determine their adoption within the IoBNT framework. Currently, this aspect of NNs in IoBNT is explored less than their functional applications.

As summarized in Table V, the literature remains scarce and focuses on a limited set of NN architectures, mostly feedforward NNs. Preliminary research interprets the operation of NNs using classical methods, including threshold, slope detectors, and zero-forcing precoders. In the context of MC, XAI remains nascent, and the interpretability of NN modules across other layers of IoBNT networks remains an open research challenge.

Lastly, a particularly promising direction is to incorporate physical laws directly into the training process via physics-informed neural networks (PINNs). Unlike purely data-driven approaches, PINNs can embed the governing partial differential equations (PDEs) of molecular transport (e.g., Fick's diffusion law, reaction-diffusion kinetics, or fluid dynamics in vesicle-mediated signaling) as additional loss terms within the training objective. Not only do these models exhibit better generalization to previously unseen conditions, but they also offer greater interpretability of the solution space, as latent layers must satisfy fundamental conservation laws and boundary conditions intrinsic to molecular transport.

VI. The Backbone of Neural Networks: Training Data

Having discussed the modeling approaches and learning methodologies for MC systems in the previous sections, it becomes apparent that their performance critically depends on the availability and quality of data. Data is the backbone of NN training, and this section provides an overview of existing MC datasets. We include datasets that have not yet been exploited in the context of NNs. Furthermore, the datasets are discussed with respect to their generation and accessibility, and some remarks are made on their documentation. Finally, current limitations and future perspectives on synthetic data generation are outlined.

Many datasets, particularly those derived from simulations and published in papers, i.e., data used for plotting, are not publicly available. This lack of accessibility undermines the reproducibility and validation of the findings, limiting the research community's ability to build on existing research. By focusing on publicly available and accessible datasets, such as [201], [202], [203], we aim to provide a comprehensive summary that researchers across domains can use.

A. General Considerations

Data can be generated based on observations of physical or virtual processes, such as wet-lab experiments or computer simulations. Data generation can be time-consuming and resource-intensive. Avoiding repeating the process, datasets shared among different research groups can (i) facilitate reproducible research and (ii) provide an abstraction layer for the actual process of interest, e.g., MC between synthetic cells,

TABLE V: Summary of XAI research in molecular communication.

XAI Method	NN-model	MC-Geometry	Focus	Description & Key findings	Ref.
LIME, ICE	Feedforward NN	Air-based free diffusion channel	Symbol detection	Interpret the NN operation as a receiver based on the identification of the most critical samples in the received sequence over MC channels. As a result, the NN is interpreted to operate as a peak detector or as a slope detector based on the ISI level of transmissions.	[197]
		2×2 MIMO in free diffusion channel			[196]
SHAP		Point transmitter - Free diffusion - Spherical receiver			[198]
SHAP, MPI		Extracellular matrix	Distance estimation	Quantifies the contribution of peak amplitude and timing features to the predicted cell-to-cell distance. The NN is interpreted as an estimator based on the peak time of received pulses.	[199]
Linear modeling	Autoencoder	Drifted channels	End-to-end learning of MC channels	Develop a surrogate model to interpret the transmitter's operation as a zero-forcing precoder.	[104]

that allows other researchers to develop models and algorithms without the need to (re-)run and observe the process.

Both these goals hinge on the *availability* and the *usability* of the datasets. Also, a thorough *documentation* of the shared data is required. These aspects, which relate primarily to data and metadata handling, will be covered later in this section. However, in determining which data to collect in the first place, the properties of the process that generates the data and the primary purpose of data collection play a key role.

The purpose of collecting data can be to validate a specific theoretical model. In this case, it is essential to control the experimental conditions as best as possible and obtain a representative set of data samples under these conditions. The main focus of data collection ito confirm or reject a specific, oftenas is the case, e.g., for NN training quantitative, hypothesis on the process, e.g., the average received SNR, that originates from the theoretical model.[21] On the other hand, data can also be used to build a model, as in NN training. In this case, data should be collected to avoid training-set imbalances and overfitting to a narrow subset of experimental conditions.

Training data can be distinguished into two types: (i) Synthetic and (ii) experimental data. In this survey, synthetic data refers to data artificially generated through algorithms or simulations that mimic real-world scenarios. Experimental data is collected from experimental samples, alongside the challenges of building environments in which various parameters can be precisely controlled.

B. Synthetic Data Generation

1) Existing Datasets

Existing datasets primarily address two environments: Pipe-like and free-diffusion channels in fluid and air-based media. For pipe-like channels, the authors in [204] introduce a cylindrical water-filled duct designed to study steady-state flow conditions across various channel lengths. The authors benchmark their simulation results against particle-based simulation (PBS) and an analytical channel modeling approach, as accessible in [205].

The above dataset has also been extended in [206] to cover not only the flow-dominated regime but also the dispersion and the mixed regimes. Furthermore, targeting healthcare applications for advanced plaque modeling [207], the work in [208] develops a dataset accessible in [209], using the computational fluid dynamics (CFD) solver OpenFOAM. A previously published dataset in [205] provides a framework for MC specific post-processing, i.e., utilizing Python to analyze the OpenFOAM simulation output and evaluate the CIR of the MC channel.

The dataset in [210] includes results from CFD simulations performed using the proprietary ANSYS software to study the transmission of airborne pathogens in turbulent airflow environments. The simulations examine the spread of pathogen-laden droplets and aerosols, "mimicking scenarios such as coughing-induced flows" [210]. An overview and summary of all simulation results, along with a detailed explanation, can be found in [211].

While the datasets discussed so far mostly contain data from fluid dynamics simulations, some other datasets reproduce time sequences as a result of processing algorithms. This is the case in [212], where a macroscale MC system—using an electric alcohol sprayer and sensor without forced airflow—is provided as Matlab files and raw measurement data. The estimated distance is produced by ML-based methods, see [39], and the Fluid Dynamics-based Distance Estimation (FDDE) algorithm, see [213].

The dataset in [214] includes, in addition to the dataset itself, the code for creating it in Matlab, which is published on a university server, making the results from [215], [216] accessible to other researchers. The authors of [215], [216] address the development of multiple-access schemes and detection strategies for diffusion-based MC, focusing on mitigating ISI and multiple-access interference (MAI). Furthermore, [215] and [216] introduce the type-spread molecule shift keying (MoSK) modulation technique, which uses additional molecule types to reduce the ISI effect. Type-spread MoSK is expanded for a multiple-access system to support multiple nanomachines. In addition, a molecular code-division multiple-access scheme is proposed in [215], [216], relying on two molecule types.

Published datasets can also describe the parameters used in a publication [217]. For example, in [218], an improved channel model for MC with a partially absorbing receiver is proposed. The partially absorbing receiver has four parameters determined by particle swarm optimization (PSO). The optimum of the

[21]The theoretical model can also be an NN, in which case the hypothesis results from the inference step.

determined parameters, as mentioned in [218], can be found in the corresponding dataset in [217].

2) Dataset Generation Tools

In addition to the published datasets, simulators and software tools are also available to generate synthetic datasets based on users' requirements and system model parameters. Generally, in some MC simulators, the medium flow is simulated explicitly, whereas other tools use analytical models to approximate or neglect it [219]. The latter set of tools focuses mainly on Monte Carlo simulation and is referred to as *flow-agnostic simulators*. In contrast, the physical fluid flow characteristics are considered in the former one, referred to as *flow-aware simulators*. Most of the tools are publicly available, except for the commercial simulators COMSOL Multiphysics, ANSYS Fluent, and Matlab, for which a license is required to utilize the full range of available functions. Further requirements also arise for the MC graphical processing unit (GPU) simulator (also known as "parallel simulation framework for nanonetworking" [220]), as it requires an Nvidia GPU supporting the Nvidia compute unified device architecture (CUDA) [220].

The flow-agnostic simulators encompass tools designed for various applications, such as bacterial MC (nanoNS3 [221], BNSim [222]), reaction-transport modeling (URDME [223]), and Brownian motion-based simulations (Smoldyn [224], [225], N3Sim [226], [227], AcCoRD [228]). Diverse computational frameworks support these simulators, including Matlab, COMSOL, and NS3. In contrast, fluid dynamics simulators used in the context of MC, such as OpenFOAM, COMSOL Multiphysics, and ANSYS Fluent, are primarily based on the concept of CFD and numerical solutions of the Navier-Stokes equation in general. Availability varies significantly, with several simulators offering freely accessible source codes (e.g., OpenFOAM, Munich microfluidic toolkit, and NS3). In contrast, others, like ANSYS Fluent and COMSOL Multiphysics, are commercially licensed. This highlights the trade-off between cost and accessibility across the simulators, with open-source tools typically targeting niche research applications and commercially available software catering to broader engineering applications.

In addition to the simulators, other existing analytical simulation codes, data generation codes, and models have been published on GitHub as repositories for code development. This is the case in [229], which models a free-diffusion MC channel and implements an NN-based detector. Three other GitHub projects have been created as part of a lecture series in which the results in [64] were reproduced.[22] Furthermore, a trained artificial neural network (ANN) model is published on Matlab Central [233] to evaluate the number of received molecules for a spherical reflecting transmitter and a spherical absorbing receiver using an ANN approach [38].

Matlab code is also developed to simulate the random walk and the signal reconstruction process at the receiver in free-diffusion channels [234], [235]. Introducing the mobile human ad hoc network (MoHANET) concept in which MC principles and analysis are applied to pathogen-laden droplets, code developments also encompass airborne pathogen transmission, integrating insights from epidemiology, biology, medicine, and fluid dynamics [236], [237]. These code files represent proof-of-concept results validated using empirical COVID-19 data from the reference [238]. Finally, the published code also accounts for the modeling of a stochastic biofilm formation model based on bacterial QS [239], [240], [241], [242].

The code in [243] includes an NN model to realize a signal sequence detector for a mobile MC system based on the Informer model in [76]. Considering a diffusion-based environment, the mobile MC system comprises a point transmitter and a spherical passive receiver [76]. The signal sequence detector computes the autocorrelation coefficient of the input sequence to determine the optimal sequence length. Numerically obtained results in [76] demonstrate that the performance of the Informer-based model is better than that of the Transformer-based model in terms of detection ability. However, the dataset is not publicly available.

In [244], a "model for generating synthetic data for a biological MC" for training an NN for the discrimination of transmitted bits is presented. Testbed measurement data from [245] is used as a benchmark for the synthetically generated data. The corresponding dataset is available upon request. Table VI reviews the existing dataset generation tools. To assess their accessibility, we introduce three complementary metrics that characterize code availability and their accesbility; see table notes.

C. Experimental Data Generation

Experimental data generation from testbeds can generally be distinguished into air-based [203], [246] and liquid-based [201], [202], [247], [248], [249], [250], [251], [252], [253], [254], [255], [256] data. As listed in [29], there are more testbeds than those mentioned here, but, to the best of the authors' knowledge, no data were made publicly available for other testbeds.

The datasets published on IEEE Dataport [201] and Zenodo [202] include fluorescence signal measurements (emission wavelength and intensities) of MoSK transmissions in a liquid-based testbed, which was proposed in the related publication in [257]. In the testbed setup, graphene quantum dots (GQDs), soluble in water and fluorescent, serves as a signaling molecule. The transmitter infuses the GQDs using injection valves, and a fluorescence-based receiver detects and decodes the fluorescence signals from blue-GQDs and cyan-GQDs, which serve as the molecules to shift. The testbed's performance is assessed based on synchronization, detection thresholds, and symbol recognition through a principal component analysis (PCA), which requires a broad dataset.

The dataset in [203] provides experimental measurements from a macroscale, single-input, single-output, air-based MC testbed. It also includes the Matlab processing code of the dataset-related publication in [197]. The dataset has been reported in various research studies in the literature, including XAI in MC channels [197], RL-based synchronization mechanisms [52], [54], and adaptive detectors [53]. The dataset in [203] was extended as a result of the research in [258]. The new associated repository, accessible in [246], records the experimental measurement data for ethanol molecules in the

[22]These code projects are (i) Sangani [230], (ii) Patel [231] (also improved the code in terms of the CIR of a passive receiver or enzymatic degradation), and (iii) Shastri in [232].

air. It also provides data sheets for the testbed components and Python code to control the sprayer, read the sensor output, and implement communication protocols.

The dataset in [247] publishes a biocompatible MC testbed that utilizes magnetic nanoparticles as information carriers, so-called superparamagnetic iron oxide nanoparticles (SPIONs). The SPIONs are injected into a constant background flow using an injection pump, cf. [259, Fig. 1]. The receiver detects the SPIONs by a change in the inductance of the nearby fluid caused by the presence of the SPIONs. This testbed is utilized to investigate channel parameters such as (among others) background flow, channel length, and channel diameter. It should be emphasized that [259] discusses the dataset in detail and explains its structure, which is a limitation in other published datasets.

The received sequence of interfacial shift keying (ISK) transmissions in a liquid-based testbed is accessible on IEEE DataPort [248] and Zenodo [249]. In ISK, the modulation of the signal exploits the effect of viscosity fingering, i.e., two miscible fluids form a (temporary) interface, given that they differ in either their viscosity or density [294]. The testbed for experimental evaluation consists of a transmitter containing an infusion pump, a six-way injection valve, and a 10-way selection valve, which allows the injection of up to ten solutions [294, Fig 1]. Fluorescent carbon nanoparticles are used as information carriers. On the receiver side, a fluorescence detector measures the system's fluorescence output, demodulating and decoding the transmitted signal.

Images obtained from the particle image velocimetry (PIV) and planar laser-induced fluorescence (PLIF) tools, which are referred to for tracking and detecting fluorescent tracers in liquid, are accessible in [250], [251]. The dataset results from the research in [295], which develops two methodologies for particle tracking and detection in liquids. Both methods are based on a laser sheet illuminating a planar section of the medium [295, Figs. 2 and 3], where fluorescent tracers, captured by a camera, serve as information carriers. The repository also includes Matlab code for further image processing to track and detect the fluorescent tracers.

The open-access dataset in [252] contains experimental measurements from a biological MC testbed [296]. The testbed utilizes *E. Coli* bacteria, which "express the light-driven proton pump *gloeorhodopsin* from *Gloeobacter violaceus*." When stimulated by external light, the bacteria act as transmitters, releasing protons into a liquid channel. The protons serve as signaling molecules, altering the system's pH, which is later detected. The repository also provides a detailed description of the dataset and comprises two zip files containing the data along with Matlab code example for processing.

In [139], a liquid-based microfluidic MC testbed is presented. The source files for [139] are accessible in [253], [297]. For transmitting the information, CSK is used; in particular, information is encoded in the concentration of sodium hydroxide in that testbed. In addition, the testbed models chemical reactions and microfluidic geometry, enabling the transmitter to shape the signal and the receiver to threshold, amplify, and detect it after propagation [139]. Therefore, the chemical reactions are based on well-known pH-based reactants, such as hydrochloric acid and sodium chloride [139]. A phosphate-buffered saline solution is used for dilutions, and a spectrometer on the receiver side detects the transmitted information. The raw data (mainly as `csv` files) in [253] refers to the plotted results in [139]. The GitHub repository in [297] contains the software for the complete automation of the testbed, including timed chemical injection using syringe pumps, measurement of the flow rate, and control of the flow rate using a proportional–integral—derivative controller, and measurement of the UV-visible spectrum.

The freely accessible dataset in [255] contains experimental measurements from long-term experiments using the closed-loop MC testbed described in [298]. The experiment was run for $125\,\mathrm{h}$, obtaining more than $250\,\mathrm{kbit}$ of transmitted data via MC. The testbed utilizes a green fluorescent protein variant "Dreiklang" (GFPD), as the information carrier. Using light of a specific wavelength, GFPD can be switched reversibly between two different states [299]. The testbed differs from other liquid-based testbeds because it is a closed-loop structure, not an end-to-end structure, e.g., by pumping a liquid from one reservoir to another. Therefore, GFPD is only injected once. The testbed contains an optical transmitter for writing information, an optical eraser for erasing information, and a receiver for reading the fluorescence state of the GFPD [298]. In [255], the raw measurement dataset (as a zip folder containing `csv` and `json` files and as a SQLite database) is published, corresponding to the plots in the publication [300]. In addition, the Python code for synchronization and detection is available in a GitHub repository [254], which includes instructions for reading the SQLite database.

The Zenodo dataset in [256] contains 69 experimental measurements from the study in [301], which introduces the first versatile three-dimensional *in vivo* MC testbed. Each entry includes egg and region identifiers, developmental stage, injection parameters, fluorescence intensity, estimated distributions, and corresponding error values. This dataset offers a comprehensive experimental basis for modeling and analyzing *in vivo* MC channels.

D. Discussion

The datasets mentioned above exhibit different levels of accessibility, completeness, and documentation. As a reader's guide, we develop a traffic-light system in Table VII in which green represents the highest level of dataset completeness and red represents the least along the following dimensions:

- Reproducibility (only considered for synthetic datasets): Green dots indicate that the source code and its documentation are fully accessible. Yellow dots indicate that the dataset documentation is missing or that access to the source code is unavailable. The red dots indicate that all of the above are missing.
- Representativeness (only for experimental datasets): Evaluated by comparing the number of replicates (N) of single experiments to the maximum number of repetitions over all experimental datasets. For the considered set of experimental datasets, a maximum number of $N = 69$ [256] replicates applies so that for the ranges between $N = 1$ and $N = 23$ replicates a red dot, between $N = 24$ and

TABLE VI: Overview of data-generating simulators, alphabetically sorted. URDME and MMFT denote the Unstructured Reaction-Diffusion Master Equation and the Munich MicroFluidic Toolkit, respectively.

Category	Simulator and dependence	Application area	Source code	Open-source	Availability	Accessibility
Flow agnostic	AcCoRD [228]	Microscopic and mesoscopic diffusion-based MC	GitHub [260]	🟢	🟢	🟢
	BiNS [261] & BiNS2 [262]	Diffusion-based MC in blood vessels		🔴	🟡	🔴
	BioNetGen [263]	Biochemical systems	Website/manual [264]	🟢	🟢	🟢
	blood-voyager-s [118], BVS-Vis [265], MEHLISSA [266] BVS-Net [270] (ns-3 extensions)	Moving objects in human body	GitHub, including blood-voyager-s [267], BVS-Vis [268], MEHLISSA [269], BVS-Net [271]	🟢	🟢	🟢
	BNSim [222]	Bacterial networks	Website [272]	🟢	🟢	🔴
	MC GPU Simulator (Nvidia GPU supporting CUDA)	Diffusion-based MC	GitHub [220]	🟢	🟢	🟢
	MesoRD [273] (visualising results in Matlab)	Reaction-diffusion simulations	SourceForge [274]	🟢	🟢	🟢
	MolComSim	Simple active and passive MC	GitHub [275]	🟢	🟢	🟢
	MUCIN [276] (Matlab extension)	Diffusion-based MC with drift	Matlab Central File Exchange [277]	🟡	🟡	🟡
	Multicellular MC Simulator [278], [279]	(Large scale) multicellular MC scenarios	GitHub [280]	🟢	🟢	🟢
	nanoNS3 [221] (ns-3 extension)	Bacterial MC networks	Download [281]	🟢	🟢	🟢
	N3Sim [226], [227]	Diffusion-based MC	Website [282]	🟡	🟡	🔴
	N^4Sim [283]	Nervous systems and synaptic MC	GitHub [284]	🟢	🟢	🟢
	Smartcell [285]	Cellular processes	Available upon email request; initial download link not available	🟢	🟡	🟡
	Smoldyn [224]	Diffusion, membrane interactions, and molecule reactions	Website [286]	🟢	🟢	🟢
	URDME [223] (including Matlab and COMSOL interface)	Reaction-transport simulation and modeling	GitHub [287]	🟢	🟢	🟢
Flow aware	ANSYS Fluent	Wide range of fluidic scenarios		🔴	🟡	🟡
	COMSOL Multiphysics	Wide range of fluidic scenarios		🔴	🟡	🟡
	Droplet-Based Microfluidic Simulator [288], [289] (MMFT extension)	Droplet-based fluidic MC	Github [290]	🟢	🟢	🟢
	Matlab	PDEs, CFD, and particle tracking for MC and fluidic scenarios	Commercially available	🔴	🟡	🟡
	OpenFOAM (Pogona [291], [292])	Wide range of fluidic scenarios	Pogona- GitHub [293]	🟢	🟢	🟢

Notes:
• The first metric, open-source status, assesses whether the underlying source code is publicly accessible and modifiable. Tools that are fully open-source receive a green rating, partially accessible or account-restricted tools receive a yellow rating, and closed-source tools whose source code is not publicly released receive a red rating.
• The second metric, availability, evaluates how easily a simulator can be obtained through formal release channels. Freely downloadable or straightforwardly purchasable tools receive a green rating, whereas tools requiring registration, institutional affiliation, or academic request are marked yellow. Tools that are no longer supported or unavailable receive a red rating.
• Finally, the accessibility metric captures the hurdles users must overcome to download or install the software. Simulators offering direct, barrier-free downloads receive a green rating; those requiring additional steps, such as email requests or account creation, receive a yellow rating; and those with broken links, discontinued repositories, or inaccessible downloads receive a red rating.

$N = 46$ replicates a yellow dot, and between $N = 47$ and $N = 69$ replicates a green dot was assigned. The dataset in [212] is rated in yellow because it contains a total of $N = 55$ measurements; however, only $N = 5$ repetitions are conducted for each transmitter-receiver distance. The dataset in [254], [255] is an exception here, as it is the first long-term experimental MC system of its kind. The dataset size clearly stands out from the other datasets and is consequently rated green.

- Usability: Encompasses the cases "the dataset is available", "the dataset processing code is available", and "the code for plotting is available". If all three criteria are met, green follows; for two criteria (regardless of which), yellow follows; otherwise, red is applied.
- Availability: Here, we consider only two cases, either yellow (data record restricted availability, for example, behind an account wall) or green (data record freely available). Datasets that are not available are not listed, so no red is assigned here.
- Documentation: Considers the completeness of the parameters' metadata and the dataset's documentation. Green: Both aspects are fulfilled; Yellow: Only one of them is

TABLE VII: 4D dataset clustering for synthetic and experimental data. Green, yellow, and red circles represent the highest, medium, and lowest levels, respectively.

Reference	Reproducibility	Representativeness	Usability	Availability	Documentation
[209]	●	-	●	●	●
[201], [202]	-	●	●	●	●
[205]	●	-	●	●	●
[246]	-	●	●	●	●
[203]	-	●	●	●	●
[247]	-	●	●	●	●
[248], [249]	-	●	●	●	●
[250], [251]	-	●	●	●	●
[252]	-	●	●	●	●
[214]	●	-	●	●	●
[253]	-	●	●	●	●
[210]	●	-	●	●	●
[217]	●	-	●	●	●
[302]	●	-	●	●	●
[303]		●	●	●	●
[212]	-	●	●	●	●
[304]	●	-	●	●	●
[254], [255]	-	●	●	●	●
[256]	-	●	●	●	●

fulfilled; Red: None of them are fulfilled.

Available datasets for MC research are diverse and accessible through multiple platforms; however, several limitations hinder broad use, cf. Table VII. One limitation is the lack of standardization in data formats and annotations, which complicates integration and comparative analysis across datasets. Additionally, some datasets lack detailed metadata or related publications, reducing the transparency and reproducibility of the research outcomes. The sizes of certain datasets are disproportionately small, limiting their applicability for machine learning or extensive simulation studies. Another issue is platform dependence. While IEEE DataPort, GitHub, and Zenodo are widely used, access to some university-hosted datasets may be restricted or not well-documented.

Data generation tools face compatibility limitations, as many rely on specific frameworks such as Matlab or NS3, hindering reproducibility across software versions. Moreover, simulator-specific modeling assumptions restrict dataset representativeness. For example, reaction-diffusion tools such as Smoldyn or MesoRD struggle with environmental heterogeneity, whereas nanoNS3 or N3Sim may not generalize well to macroscopic systems. Availability and usability further limit adoption, since some tools require licenses, lack active maintenance, or provide insufficient documentation for novice users. Finally, dataset quality strongly impacts NN generalization, as label noise and class imbalance can cause overfitting and biased predictions, making high-quality, balanced data as critical as model architecture design [305], [306], [307].

E. Concluding Remarks and Outlook

To summarize, in recent years, data generation, documentation, and publication have received attention, addressing the need for robust training datasets and reliable research outcomes. Our survey of existing datasets also shows that eight datasets are derived from experimental testbeds. The use case for sharing experimental data is stronger than for synthetic data, primarily because wet-lab experiments are more challenging to reproduce than simulations. Additionally, dataset publication still requires more extensive and comprehensive documentation practices.

In this topic, we also identify further research directions towards the connections between testbeds and the training and deployment of NNs as follows:

1. Training in Body-like Testbeds: The pharmaceutical industry is today testing experimental results developed in organs-on-chip. As a proven tool for drug testing [312], this technology mimics organ function and fluid dynamics. As these devices are expected to become a daily occurrence in laboratories, we anticipate they will serve as testbeds for generating data and integrating NN-models.

2. Transferring from Simulators to Real-World Testbeds: A broad range of simulation tools exists for MC; on the one hand, these simulation tools are often open-source and easy to install, so researchers use them to implement their NN/ML approaches and benchmark them against analytical approaches, such as in [313]. On the other hand, the components of experimental testbeds are often limited in their computational capabilities, as seen in the testbed presented in [53], making the implementation of computationally intensive NN and ML approaches challenging. To address these limitations, recent research explores lightweight NN architectures that lower resource demands without compromising accuracy. Techniques such as model quantization [314], network pruning [315], and TinyML [316] enable efficient deployment of models on resource-constrained testbeds. However, as the MC community moves towards practical applications, NN and ML tools applied to MC simulation tools should be transferred to experimental testbeds. In this way, experimental setups will enable benchmarking, at least in pre-defined real-world scenarios, owing to the targeted parameter control of the experimental testbeds.

VII. Conclusion

This survey examines the variety of NN architectures for the functional development of IoBNT networks and provides an overview of the most recent methods for integrating them into MC links. A thorough evaluation of the reported research concludes that NN architectures are essential for handling the complex nature of MC connections, for which closed-form expressions for model-based solutions are often impractical to derive. As such, NNs are the *de facto* model for devising practical solutions within the various layers of IoBNT networks. A closer look at the reported literature reveals that most of the reported developments primarily focus on the PHY layer for decoding tasks, where the BiRNN achieves the highest performance in ISI channels and with less complexity compared to RNNs, feedforward NNs, and CNNs. Conversely, fewer works target solutions in the upper layers, and only a few methods explore the much-needed localization and detection applications in IoBNT networks. We expect further developments in NN-based methods towards application-oriented solutions for IoBNT networks. The literature is also maturing in supplying the necessary datasets for NN training, and we anticipate further progress in the usability and reproducibility dimension of datasets, along with increased data generation for in-body environments.

Furthermore, studies at the intersection of NNs and IoBNT underline two long-term research directions: (i) biocompatible

deployment of NNs at the nanoscale is still in its early stages, and nanoscale architectures beyond feedforward NNs require further attention to realize the full potential of IoBNT; and (ii) physics-informed architectures, grounded in domain knowledge, can support the development of self-explainable solutions. We conclude with the broader perspective that AI can provide a common conceptual and methodological framework for bridging biology and engineering, enabling interdisciplinary designs in which biological processes and communication systems are developed in closer alignment.

REFERENCES

[1] E. Schrödinger, *What is Life? The Physical Aspect of the Living Cell*. Cambridge University Press, 1944, p. 196.

[2] I. F. Akyildiz, M. Ghovanloo, U. Guler, T. Ozkaya-Ahmadov, A. F. Sarioglu, and B. D. Unluturk, "PANACEA: An Internet of Bio-NanoThings Application for Early Detection and Mitigation of Infectious Diseases," *IEEE Access*, vol. 8, pp. 140 512–140 523, Jan. 2020.

[3] J. Torres Gómez et al., *Dataset for Cell-to-Cell Communications*, IEEE Dataport, 2024. [Online]. Available: doi.org/10.21227/vv2r-qn28.

[4] N. Etemadi, M. Farahnak-Ghazani, H. Arjmandi, M. Mirmohseni, and M. Nasiri-Kenari, "Abnormality Detection and Localization Schemes Using Molecular Communication Systems: A Survey," *IEEE Access*, vol. 11, pp. 1761–1792, 2023.

[5] E. Lagasse and M. Levin, "Future medicine: from molecular pathways to the collective intelligence of the body," *Trends in Molecular Medicine*, vol. 29, no. 9, pp. 687–710, Sep. 2023.

[6] M. Kuscu and B. D. Unluturk, "Internet of Bio-Nano Things: A review of applications, enabling technologies and key challenges," *ITU Journal on Future and Evolving Technologies*, vol. 2, no. 3, pp. 1–24, Dec. 2021.

[7] D. Aktas et al., "Odor-Based Molecular Communications: State-of-the-Art, Vision, Challenges, and Frontier Directions," *IEEE Communications Surveys & Tutorials*, pp. 1–34, 2024.

[8] Y. Lu, R. Ni, and Q. Zhu, "Wireless Communication in Nanonetworks: Current Status, Prospect and Challenges," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 6, no. 2, pp. 71–80, Nov. 2020.

[9] J. L. Marzo, J. M. Jornet, and M. Pierobon, "Nanonetworks in Biomedical Applications," *Current Drug Targets*, vol. 20, no. 8, pp. 800–807, May 2019.

[10] K. Yang et al., "A Comprehensive Survey on Hybrid Communication in Context of Molecular Communication and Terahertz Communication for Body-Centric Nanonetworks," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 6, no. 2, pp. 107–133, Nov. 2020.

[11] B. Dong, Y. Ma, Z. Ren, and C. Lee, "Recent progress in nanoplasmonics-based integrated optical micro/nano-systems," *Journal of Physics D: Applied Physics*, vol. 53, no. 21, Mar. 2020.

[12] N. Saeed, M. H. Loukil, H. Sarieddeen, T. Y. Al-Naffouri, and M. S. Alouini, "Body-Centric Terahertz Networks: Prospects and Challenges," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 8, no. 3, pp. 138–157, Sep. 2022.

[13] S. Abadal et al., "Graphene-Based Antenna Design for Communications in the Terahertz Band," in *Nanoscale Networking and Communications Handbook*, J. R. Vacca, Ed., 1st ed., Boca Raton, FL: CRC Press, 2019, pp. 25–45.

[14] S. Abadal, C. Han, V. Petrov, L. Galluccio, I. F. Akyildiz, and J. M. Jornet, "Electromagnetic Nanonetworks Beyond 6G: From Wearable and Implantable Networks to On-Chip and Quantum Communication," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 8, pp. 2122–2142, Aug. 2024.

[15] X.-X. Yin, A. Baghai-Wadji, and Y. Zhang, "A Biomedical Perspective in Terahertz Nano-Communications—A Review," *IEEE Sensors Journal*, vol. 22, no. 10, pp. 9215–9227, May 2022.

[16] S. Canovas-Carrasco, R. Asorey-Cacheda, A.-J. Garcia-Sanchez, J. Garcia-Haro, K. Wojcik, and P. Kulakowski, "Understanding the Applicability of Terahertz Flow-Guided Nano-Networks for Medical Applications," *IEEE Access*, vol. 8, pp. 214 224–214 239, 2020.

[17] F. Lemic et al., "Survey on Terahertz Nanocommunication and Networking: A Top-Down Perspective," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1506–1543, Jun. 2021.

[18] R. C. Moioli et al., "Neurosciences and Wireless Networks: The Potential of Brain-Type Communications and Their Applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1599–1621, 2021.

[19] J. M. Jornet and A. Sangwan, "Nanonetworking in the Terahertz Band and Beyond," *IEEE Nanotechnology Magazine*, vol. 17, no. 3, pp. 21–31, Jun. 2023.

TABLE VIII: Existing datasets on MC, ordered by platform and year of publication.

	Name of the dataset	Reference	Year	Size	Data type	Related work	Platform
(1)	Dataset for Advanced Plaque Modeling for Atherosclerosis Detection using Molecular Communication	[209]	2024	∼1.5 GB	Synthetic	[308]	IEEE DataPort
(2)	Dataset of "Experimental Implementation of Molecule Shift Keying for Enhanced Molecular Communication"	[201], [202]	2023	∼300 MB	Experimental	[257]	IEEE DataPort [201], Zenodo [202]
(3)	Dataset for the Simulation of Microfluidic Molecular Communication using OpenFOAM	[205]	2023	∼40 GB	Synthetic	[204]	IEEE DataPort
(4)	Dataset for Analog Network Coding in Molecular Communications: A Practical Implementation	[246]	2023	∼0.8 MB	Experimental	[258]	IEEE DataPort
(5)	Dataset for Macroscale Molecular Communication Testbed	[203]	2023	∼0.2 MB	Experimental	[52], [53], [54], [197]	IEEE DataPort
(6)	Channel Parameter Studies with a Biocompatible Testbed for Molecular Communication	[247]	2023	∼2 MB	Experimental	[259]	IEEE DataPort
(7)	The Data Related to Interfacial Shift Keying Allows a High Information Rate in Molecular Communication	[248], [249]	2022	∼0.6 MB	Experimental	[294], [309]	IEEE DataPort [248], Zenodo [249]
(8)	Molecular Signal Tracking and Detection Methods in Fluid Dynamic Channels (+ Method and Data)	[250], [251]	2019, 2020	∼26.8 GB	Experimental	[295]	IEEE DataPort
(9)	A Molecular Communication Testbed Based on Proton Pumping Bacteria	[252]	2019	∼0.6 MB	Experimental	[296]	IEEE DataPort
(10)	Dataset in Support of the Southampton Doctoral Thesis "Type-Spread and Multiple-Access Molecular Communications"	[214]	2023	∼25 MB	Synthetic	[216]	University server
(11)	Transient Indocyanine Green Distribution Measurement and Modelling in Chorioallantoic Membrane (CAM) Model	[256]	2025	∼ 220 MB	Experimental	[301]	Zenodo
(12)	Closed-Loop Long-Term Experimental Molecular Communication System	[254], [255]	2025	∼667 MB	Experimental	[57], [298], [299], [300]	Zenodo [255], GitHub [254]
(13)	CFD Simulation Dataset for Airborne Pathogen Transmission in Turbulent Channels	[210]	2024	∼506.3 MB	Synthetic	[211], [310]	Zenodo
(14)	Real-Time Signal Processing via Chemical Reactions for a Microfluidic Molecular Communication System	[253]	2023	∼2.4 GB	Experimental	[139], [297]	Zenodo
(15)	Received Signal Modeling and BER Analysis for Molecular SISO Communications	[217]	2022	∼3.1 kB	Synthetic	[218]	Zenodo
(16)	Channel Estimation and Performance Analysis of SISO Molecular Communications	[302], [311]	2021	∼3.3 kB	Synthetic	Not specified	Zenodo
(17)	"molecular_communication"	[303]	2020	∼1.5 kB	Not specified	Not specified	Kaggle
(18)	"MCFormer: A Transformer-Based Detector for Molecular Communication with Accelerated Particle-Based Solution"	[304]	2023	∼41.8 MB	Synthetic	[68]	GitHub
(19)	"Distance-Estimation-in-Molecular-Communication"	[212]	2020	∼620 kB	Experimental	[39], [213]	GitHub

[20] S. Zafar et al., "A Systematic Review of Bio-Cyber Interface Technologies and Security Issues for Internet of Bio-Nano Things," *IEEE Access*, vol. 9, pp. 93 529–93 566, 2021.

[21] S. Qiu et al., "Review of Physical Layer Security in Molecular Internet of Nano-Things," *IEEE Transactions on NanoBioscience*, vol. 23, no. 1, pp. 91–100, Jan. 2024.

[22] I. F. Akyildiz, M. Pierobon, and S. Balasubramaniam, "Moving forward with molecular communication: from theory to human health applications [point of view]," *Proceedings of the IEEE*, vol. 107, no. 5, pp. 858–865, May 2019.

[23] 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.

[24] T. Nakano, Y. Okaie, S. Kobayashi, T. Hara, Y. Hiraoka, and T. Haraguchi, "Methods and Applications of Mobile Molecular Communication," *Proceedings of the IEEE*, vol. 107, no. 7, pp. 1442–1456, Jul. 2019.

[25] X. Huang, Y. Fang, and N. Yang, "A survey on estimation schemes in molecular communications," *Elsevier Digital Signal Processing*, vol. 124, pp. 1–13, May 2022.

[26] 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.

[27] P. Hofmann, J. A. Cabrera, R. Bassoli, M. Reisslein, and F. H. P. Fitzek, "Coding in Diffusion-Based Molecular Nanonetworks: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 16 411–16 465, 2023.

[28] B.-H. Koo, C. Lee, A. E. Pusane, T. Tugcu, and C.-B. Chae, "MIMO Operations in Molecular Communications: Theory, Prototypes, and Open Challenges," *IEEE Communications Magazine*, vol. 59, no. 9, pp. 98–104, Sep. 2021.

[29] S. Lotter et al., "Experimental Research in Synthetic Molecular Communications – Part I," *IEEE Nanotechnology Magazine*, vol. 17, no. 3, pp. 42–53, Jun. 2023.

[30] S. Lotter et al., "Experimental Research in Synthetic Molecular Communications – Part II," *IEEE Nanotechnology Magazine*, vol. 17, no. 3, pp. 54–65, Jun. 2023.

[31] 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.

[32] A.-A. A. Boulogeorgos, S. E. Trevlakis, S. A. Tegos, V. K. Papanikolaou, and G. K. Karagiannidis, "Machine Learning in Nano-Scale Biomedical Engineering," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 7, no. 1, pp. 10–39, Mar. 2021.

[33] R. T. Nagipogu, D. Fu, and J. H. Reif, "A survey on molecular-scale learning systems with relevance to DNA computing," *Nanoscale*, vol. 15, no. 17, pp. 7676–7694, 2023.

[34] A. Rizwan et al., "A Review on the Role of Nano-Communication in Future Healthcare Systems: A Big Data Analytics Perspective," *IEEE Access*, vol. 6, pp. 41 903–41 920, May 2018.

[35] P. L. Gentili, M. P. Zurlo, and P. Stano, "Neuromorphic engineering in wetware: the state of the art and its perspectives," *Frontiers in Neuroscience*, vol. 18, pp. 1–6, Sep. 2024.

[36] D. Bi, A. Almpanis, A. Noel, Y. Deng, and R. Schober, "A Survey of Molecular Communication in Cell Biology: Establishing a New Hierarchy for Interdisciplinary Applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1494–1545, 2021.

[37] M. Egan et al., "Toward Interdisciplinary Synergies in Molecular Communications: Perspectives from Synthetic Biology, Nanotechnology, Communications Engineering and Philosophy of Science," *Life*, vol. 13, no. 1, pp. 1–14, Jan. 2023.

[38] H. B. Yilmaz, C. Lee, Y. J. Cho, and C.-B. Chae, "A machine learning approach to model the received signal in molecular communications," in *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom 2017)*, Istanbul, Turkey: IEEE, Jun. 2017, pp. 1–5.

[39] F. Güleç and B. Atakan, "Distance estimation methods for a practical macroscale molecular communication system," *Elsevier Nano Communication Networks*, vol. 24, pp. 1–15, May 2020.

[40] C. Lee, H. B. Yilmaz, C.-B. Chae, N. Farsad, and A. Goldsmith, "Machine learning based channel modeling for molecular MIMO communications," in *18th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2017)*, Sapporo, Japan: IEEE, Jul. 2017, pp. 1–5.

[41] H. U. Ozdemir, H. I. Orhan, M. Turan, B. Buyuktas, and H. B. Yilmaz, "Estimating Capture Probabilities for Complex Topologies in 2D Molecular Communication via Diffusion Channel using Artificial Neural Networks," in *9th IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom 2021)*, Bucharest, Romania: IEEE, May 2021, pp. 1–6.

[42] H. U. Ozdemir, H. I. Orhan, M. Turan, B. Büyüktaş, and H. B. Yilmaz, "Estimating channel coefficients for complex topologies in 3D diffusion channel using artificial neural networks," *Elsevier Nano Communication Networks*, vol. 42, pp. 1–12, Dec. 2024.

[43] Z. Cheng, M. Chen, H. Liu, M. Xia, and W. Gong, "Channel modeling for diffusion-based molecular MIMO communications using deep learning," *Elsevier Nano Communication Networks*, vol. 42, pp. 1–9, Dec. 2024.

[44] M. Damrath, B. Yilmaz, C.-B. Chae, and P. A. Hoeher, "Array Gain Analysis in Molecular MIMO Communications," *IEEE Access*, vol. 6, pp. 61 091–61 102, 2018.

[45] Z. Cheng, H. Liu, Z. Xu, J. Li, and K. Chi, "Deep Learning-Based Estimation of Emission Time and Arrival Time in Diffusive Multi-Receiver Molecular Communication," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 11, no. 2, pp. 257–268, Jun. 2025.

[46] E. Ozbey, Y. K. Cicekdag, and H. B. Yilmaz, "Artificial neural network based misorientation correction in molecular 4x4 MIMO systems," *Elsevier Nano Communication Networks*, vol. 42, pp. 1–9, Dec. 2024.

[47] M. Zoofaghari, F. Pappalardo, M. Damrath, and I. Balasingham, "Modeling Extracellular Vesicles-Mediated Interactions of Cells in the Tumor Microenvironment," *IEEE Transactions on NanoBioscience*, vol. 23, no. 1, pp. 71–80, Jan. 2024.

[48] V. Jamali, A. Ahmadzadeh, and R. Schober, "Symbol Synchronization for Diffusion-Based Molecular Communications," *IEEE Transactions on NanoBioscience*, vol. 16, no. 8, pp. 873–887, Dec. 2017.

[49] M. Mukherjee, H. B. Yilmaz, B. B. Bhowmik, J. Lloret, and Y. Lv, "Synchronization for Diffusion-Based Molecular Communication Systems via Faster Molecules," in *IEEE International Conference on Communications (ICC 2019)*, Shanghai, China: IEEE, May 2019, pp. 1–5.

[50] H. ShahMohammadian, G. G. Messier, and S. Magierowski, "Blind Synchronization in Diffusion-Based Molecular Communication Channels," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2156–2159, Nov. 2013.

[51] N. C. Luong et al., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[52] L. Y. Debus, P. Hofmann, J. Torres Gómez, F. H. P. Fitzek, and F. Dressler, "Reinforcement Learning-based Receiver for Molecular Communication with Mobility," in *IEEE Global Communications Conference (GLOBECOM 2023)*, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 558–564.

[53] P. Hofmann, J. Torres Gómez, F. Dressler, and F. H. P. Fitzek, "Testbed-based Receiver Optimization for SISO Molecular Communication Channels," in *5th IEEE International Balkan Conference Communications and Networking (BalkanCom 2022)*, Sarajevo, Bosnia and Herzegovina: IEEE, Aug. 2022, pp. 120–125.

[54] L. Y. Debus, P. Hofmann, J. Torres Gómez, F. H. P. Fitzek, and F. Dressler, "Synchronized Relaying in Molecular Communication: An AI-based Approach using a Mobile Testbed Setup," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 3, pp. 470–475, Sep. 2024.

[55] D. Casaleiro, N. Souto, and J. C. Silva, "Synchronisation and Detection in Molecular Communication using a Deep-Learning-based Approach," *IEEE Access*, pp. 192 539–192 553, 2024.

[56] S. Szott et al., "Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1843–1893, Jul. 2022.

[57] L. Brand et al., "Media Modulation Based Molecular Communication," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7207–7223, Nov. 2022.

[58] T. Brakemann et al., "A Reversibly Photoswitchable GFP-like Protein with Fluorescence Excitation Decoupled from Switching," *Nature Biotechnology*, vol. 29, no. 10, pp. 942–947, Oct. 2011.

[59] J. Schulman, R. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv, cs.LG, Jul. 2017, pp. 1–12.

[60] M. Andrychowicz et al., "What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study," arXiv, cs.LG 2006.05990, Jun. 2020.

[61] 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.

[62] L. Y. Debus et al., "Blood Makes a Difference: Experimental Evaluation of Molecular Communication in Different Fluids," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 11, no. 4, pp. 493–499, Dec. 2025.

[63] N. Farsad, W. Guo, and A. W. Eckford, "Tabletop Molecular Communication: Text Messages through Chemical Signals," *PLoS ONE*, vol. 8, no. 12, pp. 1–13, Dec. 2013.

[64] 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.

[65] G. H. Alshammri, W. K. M. Ahmed, and V. B. Lawrence, "Adaptive Batch Training Rule-Based Detection Scheme for On-OFF-Keying Diffusion-Based Molecular Communications," in *13th IEEE Nanotechnology Materials and Devices Conference (NMDC 2018)*, Portland, OR: IEEE, Oct. 2018, pp. 1–4.

[66] U. K. Agrawal, A. K. Shrivastava, D. Das, and R. Mahapatra, "Neural Network Detector in Mobile Molecular Communication for Fast Varying Channels," in *International Conference on Connected Systems and Intelligence (CSI 2022)*, Trivandrum, India: IEEE, Aug. 2022, pp. 1–5.

[67] S. Sharma, D. Dixit, and K. Deka, "Deep Learning based Symbol Detection for Molecular Communications," in *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS 2020)*, New Delhi, India: IEEE, Dec. 2020, pp. 1–6.

[68] X. Lu, C. Bai, A. Zhu, Y. Zhu, and K. Wang, "MCFormer: A Transformer-Based Detector for Molecular Communication With Accelerated Particle-Based Solution," *IEEE Communications Letters*, vol. 27, no. 10, pp. 2837–2841, Oct. 2023.

[69] S.-J. Kim, P. Singh, and S.-Y. Jung, "A machine learning-based concentration-encoded molecular communication system," *Elsevier Nano Communication Networks*, vol. 35, pp. 1–12, Mar. 2023.

[70] O. T. Baydas, O. Cetinkaya, and O. B. Akan, "Estimation and Detection for Molecular MIMO Communications in the Internet of Bio-Nano Things," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 1, pp. 106–110, Mar. 2023.

[71] Z. Cheng, Z. Zhang, and J. Sun, "Signal Detection of Cooperative Multi-Hop Mobile Molecular Communication via Diffusion," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 1, pp. 101–111, Mar. 2024.

[72] O. Kara, G. Yaylali, A. E. Pusane, and T. Tugcu, "Molecular index modulation using convolutional neural networks," *Elsevier Nano Communication Networks*, vol. 34, pp. 1–8, Dec. 2022.

[73] A. K. Shrivastava, D. Das, and R. Mahapatra, "Performance Evaluation of Mobile Molecular Communication System Using Neural Network Detector," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1776–1779, Aug. 2021.

[74] A. K. Shrivastava, D. Das, R. Mahapatra, and N. Varshney, "Scaled Conjugate Gradient Algorithm for Neural Network Detector in Mobile Molecular Communication," in *IEEE Global Communications Conference (GLOBECOM 2021)*, Madrid, Spain: IEEE, Dec. 2021, pp. 1–6.

[75] Z. Cheng, Z. Zhang, J. Jiang, and J. Sun, "Signal Detection of Mobile Multi-user Molecular Communication System Using Transformer-Based Model," in *8th International Conference on Computer and Communication Systems (ICCCS 2023)*, Guangzhou, China: IEEE, Apr. 2023, pp. 85–90.

[76] Z. Cheng, Z. Zhang, X. Jin, W. Gong, and K. Chi, "An Informer-Based Signal Sequence Detector for Mobile Molecular Communication," *IEEE Communications Letters*, vol. 28, no. 6, pp. 1397–1401, Jun. 2024.

[77] D. Kosanetzki, O. Keszöcze, L. Kroll, J. Thiem, and J. Kirchner, "Demodulation with Deep Neural Networks for Time-Dependent Flow-Based Molecular Communication Channels," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2025)*, Barcelona, Spain: IEEE, May 2025, pp. 1–6.

[78] X. Qian and M. Di Renzo, "Receiver Design in Molecular Communications: An Approach Based on Artificial Neural Networks," in *15th IEEE International Symposium on Wireless Communication Systems (ISWCS 2018)*, Lisbon, Portugal: IEEE, Aug. 2018, pp. 1–5.

[79] T. Agrawal, *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*. Apress, 2021, p. 177.

[80] L. Chen and L. Sun, "Self-Attention-Based Real-Time Signal Detector for Communication Systems With Unknown Channel Models," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2639–2643, Aug. 2021.

[81] N. Farsad and A. Goldsmith, "Detection Algorithms for Communication Systems Using Deep Learning," arXiv, cs.LG, Jul. 2017, pp. 1–10.

[82] N. Farsad and A. Goldsmith, "Sliding Bidirectional Recurrent Neural Networks for Sequence Detection in Communication Systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, Calgary, Canada: IEEE, Apr. 2018, pp. 2331–2335.

[83] 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.

[84] B. H. Koo, C. Lee, H. B. Yilmaz, N. Farsad, A. W. Eckford, and C. B. Chae, "Molecular MIMO: From Theory to Prototype," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 600–614, Mar. 2016.

[85] B.-H. Koo, H. J. Kim, J.-Y. Kwon, and C.-B. Chae, "Deep Learning-based Human Implantable Nano Molecular Communications," in *IEEE International Conference on Communications (ICC 2020)*, Virtual Conference: IEEE, Jun. 2020, pp. 1–7.

[86] J. Wang, D. Hu, and H. Hassanieh, "Understanding and Embracing the Complexities of the Molecular Communication Channel in Liquids," in *26th ACM International Conference on Mobile Computing and Networking (MobiCom 2020)*, Virtual Conference, Sep. 2020.

[87] M. Bartunik, O. Keszocze, B. Schiller, and J. Kirchner, "Using Deep Learning to Demodulate Transmissions in Molecular Communication," in *16th IEEE International Symposium on Medical Information and Communication Technology (ISMICT 2022)*, Lincoln, NE: IEEE, May 2022, pp. 1–6.

[88] M. Bartunik, J. Kirchner, and O. Keszocze, "Artificial intelligence for molecular communication," *it - Information Technology (itIT)*, vol. 65, no. 4-5, pp. 155–163, Aug. 2023.

[89] F. Vakilipoor, D. Scazzoli, F. Ratti, G. Scalia, and M. Magarini, "Hybrid deep learning-based feature-augmented detection for molecular communication systems," in *9th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2022)*, Barcelona, Spain: ACM, Oct. 2022, pp. 1–6.

[90] C. Bai, A. Zhu, X. Lu, Y. Zhu, and K. Wang, "Temporal Convolutional Network-Based Signal Detection for Magnetotactic Bacteria Communication System," *IEEE Transactions on NanoBioscience*, vol. 22, no. 4, pp. 943–955, Oct. 2023.

[91] A. Vaswani et al., "Attention is all you need," in *31st International Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA: Curran Associates Inc., Dec. 2017, pp. 6000–6010.

[92] N. Farsad and A. Goldsmith, "Detection Over Rapidly Changing Communication Channels Using Deep Learning," in *52nd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA: IEEE, Oct. 2018, pp. 1–5.

[93] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv, cs.LG 1412.6980, Dec. 2014, pp. 1–15.

[94] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv, cs.CL 1409.0473, May 2014.

[95] C. Xiang, Y. Zhang, Y. Huang, W. Tan, X. Chen, and M. Wen, "Hybrid Recurrent Neural Network for Signal-Dependent Noise Suppression in Molecular Communication," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 11, no. 2, pp. 283–291, Jun. 2025.

[96] S. Mohamed, D. Jiang, A. R. Junejo, and D. C. Zuo, "Model-Based: End-to-End Molecular Communication System Through Deep Reinforcement Learning Auto Encoder," *IEEE Access*, vol. 7, pp. 70 279–70 286, 2019.

[97] R. Khanzadeh, S. Angerbauer, F. Enzenhofer, A. Springer, and W. Haselmayr, "Towards End-to-End Learning for Salinity-based Molecular Communication," in *7th Workshop on Molecular Communications (WMC 2023)*, Erlangen, Germany, Apr. 2023, pp. 1–2.

[98] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.

[99] Y. C. Eldar, A. Goldsmith, D. Gündüz, and H. V. Poor, *Machine Learning and Wireless Communications*. Cambridge University Press, 2022.

[100] R. Khanzadeh, S. Angerbauer, A. Springer, and W. Haselmayr, "End-to-End Learning of Communication Systems with Novel Data-Efficient IIR Channel Identification," in *57th Asilomar Conference on Signals,*

Systems, and Computers, Pacific Grove, CA: IEEE, Oct. 2023, pp. 40–46.

[101] S. Angerbauer, R. Khanzadeh, F. Enzenhofer, A. Springer, and W. Haselmayr, "Towards Asymmetric Auto-Encoders for the IoBNT," in *10th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2023)*, Coventry, United Kingdom: ACM, Sep. 2023, pp. 166–167.

[102] S. Angerbauer et al., "Salinity-Based Molecular Communication in Microfluidic Channels," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 2, pp. 191–206, Jun. 2023.

[103] R. Khanzadeh, S. Angerbauer, A. Springer, and W. Haselmayr, "End-to-End Learning for Time-Varying Non-Binary Molecular Communications," in *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, IEEE, 2025, pp. 1–6.

[104] R. Khanzadeh et al., "Explainable Asymmetric Auto-Encoder for End-to-End Learning of IoBNT Communications," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden: IEEE, May 2024, pp. 412–418.

[105] Z. Cheng, J. Sun, Z. Zhang, P. Hu, and K. Chi, "Channel Modeling and Optimal Released Molecules for Mobile Molecular MIMO Communications Among Bionanosensors," *IEEE Sensors Journal*, vol. 23, no. 19, pp. 22 139–22 152, Oct. 2023.

[106] R. Khanzadeh, S. Angerbauer, J. Torres Gómez, A. Springer, F. Dressler, and W. Haselmayr, "QL-based Adaptive Transceivers for the IoBNT Communications," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 3, pp. 476–480, Sep. 2024.

[107] Z. Cheng, J. Yan, J. Sun, S. Zhang, and K. Chi, "Resource Allocation Optimization in Mobile Multiuser Molecular Communication by Deep Neural Network," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 3, pp. 409–421, Sep. 2024.

[108] J. Torres Gómez, A. Kuestner, J. Simonjan, B. D. Unluturk, and F. Dressler, "Nanosensor Location Estimation in the Human Circulatory System using Machine Learning," *IEEE Transactions on Nanotechnology*, vol. 21, pp. 663–673, Oct. 2022.

[109] J. Torres Gómez et al., "DNA-Based Nanonetwork for Abnormality Detection and Localization in the Human Body," *IEEE Transactions on Nanotechnology*, vol. 23, pp. 794–808, Nov. 2024.

[110] J. Torres Gómez, A. Kuestner, K. Pitke, J. Simonjan, B. D. Unluturk, and F. Dressler, "A Machine Learning Approach for Abnormality Detection in Blood Vessels via Mobile Nanosensors," in *19th ACM Conference on Embedded Networked Sensor Systems (SenSys 2021), 2nd ACM International Workshop on Nanoscale Computing, Communication, and Applications (NanoCoCoA 2021)*, Coimbra, Portugal: ACM, Nov. 2021, pp. 596–602.

[111] P. Galván, F. Lemic, G. Calvo, S. Abadal, and X. Costa-Pérez, "Tailoring Graph Neural Network-based Flow-guided Localization to Individual Bloodstreams and Activities," in *11th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2024)*, Milan, Italy: ACM, Oct. 2024, pp. 109–115.

[112] X. Jin, Z. Cheng, M. Chen, H. Liu, W. Gong, and K. Chi, "Transformer-Based Receiver Localization in Vessel-Like and Flow-Induced Molecular Communication via Diffusion," *IEEE Communications Letters*, vol. 28, no. 10, pp. 2283–2287, Oct. 2024.

[113] G. C. Bartra et al., "Graph Neural Networks as an Enabler of Terahertz-Based Flow-Guided Nanoscale Localization Over Highly Erroneous Raw Data," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 8, pp. 1992–2008, Aug. 2024.

[114] M. L. Hube, F. Lemic, E. Shitiri, G. C. Bartra, S. Abadal, and X. Costa-Pérez, "Set Transformer Architectures and Synthetic Data Generation for Flow-Guided Nanoscale Localization," arXiv, cs.ET 2508.16200, Aug. 2025.

[115] O. D. Kose, M. C. Gursoy, M. Saraclar, A. E. Pusane, and T. Tugcu, "Machine Learning-Based Silent Entity Localization Using Molecular Diffusion," *IEEE Communications Letters*, vol. 24, no. 4, pp. 807–810, Apr. 2020.

[116] Z. Cheng, H. Liu, J. Zheng, W. Gong, and K. Chi, "Localizing and Tracking the Transmitter Bionanosensor in Mobile Molecular Communication by Deep Learning," *IEEE Sensors Journal*, vol. 25, no. 7, pp. 10 583–10 593, Apr. 2025.

[117] J. Torres Gómez, R. Wendt, A. Kuestner, K. Pitke, L. Stratmann, and F. Dressler, "Markov Model for the Flow of Nanobots in the Human Circulatory System," in *8th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2021)*, Virtual Conference: ACM, Sep. 2021, pp. 1–7.

[118] R. Geyer, M. Stelzner, F. Büther, and S. Ebers, "BloodVoyagerS: Simulation of the Work Environment of Medical Nanobots," in *5th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2018)*, Reykjavík, Iceland: ACM, Sep. 2018, 5:1–5:6.

[119] A. B. López et al., "Toward Standardized Performance Evaluation of Flow-guided Nanoscale Localization," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 11, no. 1, pp. 116–127, Mar. 2025.

[120] S. N. Solak and M. Oner, "Neural Network Based Decision Fusion for Abnormality Detection via Molecular Communications," in *IEEE Workshop on Signal Processing Systems (SiPS)*, Coimbra, Portugal: IEEE, Oct. 2020, pp. 1–5.

[121] S. N. Solak and M. Oner, "RNN based abnormality detection with nanoscale sensor networks using molecular communications," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020)*, Virtual Conference: ACM, Sep. 2020, pp. 1–6.

[122] T. C. Mai, M. Egan, T. Q. Duong, and M. Di Renzo, "Event Detection in Molecular Communication Networks With Anomalous Diffusion," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1249–1252, Jun. 2017.

[123] S. Mohamed, J. Dong, S. M. A. El-Atty, and M. A. Eissa, "Bio-Cyber Interface Parameter Estimation with Neural Network for the Internet of Bio-Nano Things," *Wireless Personal Communications: An International Journal*, vol. 123, no. 2, pp. 1245–1263, Sep. 2021.

[124] Q. Ma et al., "A Survey on Time-Series Pre-Trained Models," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2024.

[125] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, vol. 3, no. 7, pp. 1–28, Jul. 2018.

[126] X. Chen, L. Yang, J. Duan, and G. E. Karniadakis, "Solving Inverse Stochastic Problems from Discrete Particle Observations Using the Fokker–Planck Equation and Physics-Informed Neural Networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 3, B811–B830, Jan. 2021.

[127] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[128] R. Mosayebi, A. Ahmadzadeh, W. Wicke, V. Jamali, R. Schober, and M. Nasiri-Kenari, "Early Cancer Detection in Blood Vessels Using Mobile Nanosensors," *IEEE Transactions on NanoBioscience*, vol. 18, no. 4, pp. 103–116, Oct. 2019.

[129] L. Felicetti, M. Femminella, G. Reali, T. Nakano, and A. V. Vasilakos, "TCP-Like Molecular Communications," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 12, pp. 2354–2367, Dec. 2014.

[130] L. Lai, H. El Gamal, H. Jiang, and H. V. Poor, "Cognitive Medium Access: Exploration, Exploitation, and Competition," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 239–253, Feb. 2011.

[131] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[132] J. Torres Gómez, N. Spicher, J. L. González Rios, and F. Dressler, "Fine-tune Circuit Representation of Human Vessels through Reinforcement Learning: A Novel Digital Twin Approach for Hemodynamics," in *10th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2023)*, Coventry, United Kingdom: ACM, Sep. 2023, pp. 46–52.

[133] A. Halužan Vasle and M. Moškon, "Synthetic biological neural networks: From current implementations to future perspectives," *Biosystems*, vol. 237, pp. 1–11, Mar. 2024.

[134] A. Agiza, S. Marriott, J. K. Rosenstein, E. Kim, and S. Reda, "pH-Controlled enzymatic computing for digital circuits and neural networks," *Physical Chemistry Chemical Physics*, vol. 26, no. 31, pp. 20 898–20 907, 2024.

[135] B. Wang, M. Barahona, and M. Buck, "A modular cell-based biosensor using engineered genetic logic circuits to detect and integrate multiple environmental signals," *Biosensors and Bioelectronics*, vol. 40, no. 1, pp. 368–376, Feb. 2013.

[136] A. Hjelmfelt, E. D. Weinberger, and J. Ross, "Chemical implementation of neural networks and Turing machines," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 88, no. 24, pp. 10 983–10 987, Dec. 1991.

[137] A. O. Bicen and I. F. Akyildiz, "System-Theoretic Analysis and Least-Squares Design of Microfluidic Channels for Flow-Induced Molecular Communication," *IEEE Transactions on Signal Processing*, vol. 61, no. 20, pp. 5000–5013, Oct. 2013.

[138] D. Bi, Y. Deng, M. Pierobon, and A. Nallanathan, "Chemical Reactions-Based Microfluidic Transmitter and Receiver Design for Molecular Communication," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5590–5605, Sep. 2020.

[139] V. Walter, D. Bi, A. Salehi-Reyhani, and Y. Deng, "Real-time signal processing via chemical reactions for a microfluidic molecular communication system," *Nature Communications*, vol. 14, no. 1, pp. 1–14, 2023.

[140] A. Amerizadeh et al., "Bacterial Receiver Prototype for Molecular Communication Using Rhamnose Operon in a Microfluidic Environment," *IEEE Transactions on NanoBioscience*, vol. 20, no. 4, pp. 426–435, Oct. 2021.

[141] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A Survey of Optimization Methods From a Machine Learning Perspective," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020.

[142] M. Kuscu, H. Ramezani, E. Dinc, S. Akhavan, and O. B. Akan, "Fabrication and microfluidic analysis of graphene-based molecular communication receiver for Internet of Nano Things (IoNT)," *Scientific Reports*, vol. 11, no. 1, pp. 1–20, Oct. 2021.

[143] A. Abdali and M. Kuscu, "Frequency-Domain Model of Microfluidic Molecular Communication Channels With Graphene BioFET-Based Receivers," *IEEE Transactions on Communications*, vol. 72, no. 8, pp. 4564–4576, Aug. 2024.

[144] J. Perera et al., "Wet-Neuromorphic Computing: A New Paradigm for Biological Artificial Intelligence," *IEEE Intelligent Systems*, pp. 1–7, 2025.

[145] S. Somathilaka, S. Balasubramaniam, and D. P. Martins, "Analyzing Wet-Neuromorphic Computing Using Bacterial Gene Regulatory Neural Networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 13, no. 3, pp. 902–918, Jul. 2025.

[146] S. Somathilaka, H. Elayan, J. T. Atkinson, J. M. Jornet, and S. Balasubramaniam, "The Internet of Biofilm Living AI Devices," *IEEE Communications Magazine*, pp. 1–7, 2025.

[147] H. Cai et al., "Brain organoid reservoir computing for artificial intelligence," *Nature Electronics*, vol. 6, no. 12, pp. 1032–1039, Dec. 2023.

[148] B. J. Kagan et al., "*In vitro* neurons learn and exhibit sentience when embodied in a simulated game-world," *Neuron*, vol. 110, no. 23, pp. 3952–3969, Dec. 2022.

[149] O. Purcell and T. K. Lu, "Synthetic analog and digital circuits for cellular computation and memory," *Current Opinion in Biotechnology*, vol. 29, pp. 146–155, Oct. 2014.

[150] S. Balasubramaniam, S. Somathilaka, S. Sun, A. Ratwatte, and M. Pierobon, "Realizing Molecular Machine Learning Through Communications for Biological AI," *IEEE Nanotechnology Magazine*, vol. 17, no. 3, pp. 10–20, Jun. 2023.

[151] R. Daniel, J. R. Rubens, R. Sarpeshkar, and T. K. Lu, "Synthetic analog computation in living cells," *Nature*, vol. 497, no. 7451, pp. 619–623, May 2013.

[152] S. S. Somathilaka, S. Balasubramaniam, D. P. Martins, and X. Li, "Revealing gene regulation-based neural network computing in bacteria," *Biophysical Reports*, vol. 3, no. 3, pp. 1–21, Sep. 2023.

[153] S. Somathilaka, A. Ratwatte, S. Balasubramaniam, M. C. Vuran, W. Srisa-an, and P. Liò, "Wet TinyML: Chemical Neural Network Using Gene Regulation and Cell Plasticity," arXiv, cs.NE, Mar. 2024, pp. 1–7.

[154] J. Kim, J. J. Hopfield, and E. Winfree, "Neural network computation by in vitro transcriptional circuits," in *18th International Conference on Neural Information Processing Systems (NIPS 2004)*, Vancouver, Canada: MIT Press, Dec. 2004, pp. 681–688.

[155] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades," *Nature*, vol. 475, no. 7356, pp. 368–372, Jul. 2011.

[156] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 107, no. 12, pp. 5393–5398, Mar. 2010.

[157] K. Sarkar, D. Bonnerjee, R. Srivastava, and S. Bagh, "A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing," *Chemical Science*, vol. 12, no. 48, pp. 15 821–15 832, 2021.

[158] X. Li, L. Rizik, V. Kravchik, M. Khoury, N. Korin, and R. Daniel, "Synthetic neural-like computing in microbial consortia for pattern recognition," *Nature Communications*, vol. 12, no. 1, pp. 1–12, May 2021.

[159] L. Smirnova et al., "Organoid intelligence (OI): the new frontier in biocomputing and intelligence-in-a-dish," *Frontiers in Neuroscience*, vol. 1, pp. 1–23, Feb. 2023.

[160] A. Agiza et al., "Digital circuits and neural networks based on acid-base chemistry implemented by robotic fluid handling," *Nature Communications*, vol. 14, no. 1, pp. 1–9, Jan. 2023.

[161] A. Pandi et al., "Metabolic perceptrons for neural computing in biological systems," *Nature Communications*, vol. 10, no. 1, pp. 1–13, Aug. 2019.

[162] R. Sawlekar and G. Nikolakopoulos, "A Survey of DNA-based Computing Devices and their Applications," in *European Control Conference (ECC 2021)*, Delft, Netherlands: IEEE, Jun. 2021, pp. 769–774.

[163] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, Jul. 2018.

[164] K. R. Rodriguez, N. Sarraf, and L. Qian, "A Loser-Take-All DNA Circuit," *ACS Synthetic Biology*, vol. 10, no. 11, pp. 2878–2885, Oct. 2021.

[165] R. T. Nagipogu and J. H. Reif, "NeuralCRNs: A Natural Implementation of Learning in Chemical Reaction Networks," arXiv, cs.ET, Aug. 2024, pp. 1–42.

[166] S. Angerbauer, F. Enzenhofer, T. Pankratz, M. Hamidovic, A. Springer, and W. Haselmayr, "Novel Nano-Scale Computing Unit for the IoBNT: Concept and Practical Considerations," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, pp. 549–565, 2024.

[167] S. Angerbauer, N. Tuccitto, G. T. Sfrazzetto, R. Santonocito, and W. Haselmayr, "Investigation of Different Chemical Realizations for Molecular Matrix Multiplications," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 3, pp. 464–469, Sep. 2024.

[168] S. Angerbauer, W. Haselmayr, F. Enzenhofer, T. Pankratz, R. Khanzadeh, and A. Springer, "Molecular Nano Neural Networks (M3N):In-Body Intelligence for the IoBNT," TechRxiv, preprint, Oct. 2023, pp. 1–7.

[169] M. Uzun, K. B. Ikiz, and M. Kuscu, "Molecular Communication Channel as a Physical Reservoir Computer," arXiv, cs.ET, Apr. 2025, pp. 1–8.

[170] S. Stepney, "Physical reservoir computing: a tutorial," *Natural Computing*, vol. 23, no. 4, pp. 665–685, Nov. 2024.

[171] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, 3rd ed. Boston, MA: McGraw-Hill, 2009, p. 934.

[172] Z. W. El-Hajj and E. B. Newman, "How much territory can a single E. coli cell control?" *Frontiers in Microbiology*, vol. 6, pp. 1–12, Apr. 2015.

[173] M. Prakash and N. Gershenfeld, "Microfluidic Bubble Logic," *Science*, vol. 315, no. 5813, pp. 832–835, Feb. 2007.

[174] European Commission, "Horizon Europe Work Programme 2023-2025: 13. General Annexes," European Commission, Decision C(2025) 2779, May 2025, Annex G: Legal and financial set-up of the grant agreements. [Online]. Available: https://ec.europa.eu.

[175] F.-L. A. Lau et al., "Using Off-the-Shelf Biosensors to Implement Gateways for Alarm-System Nanonetworks," *Elsevier Nano Communication Networks*, vol. 45, p. 100 584, Sep. 2025.

[176] L. Rizik, L. Danial, M. Habib, R. Weiss, and R. Daniel, "Synthetic neuromorphic computing in living cells," *Nature Communications*, vol. 13, no. 1, pp. 1–17, Sep. 2022.

[177] 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.

[178] S. Angerbauer, F. Enzenhofer, H. Gattringer, A. Springer, and W. Haselmayr, "A Molecular Analog-to-Digital Converter," TechRxiv, cs, Jun. 2024, pp. 1–7.

[179] B. A. Holt and G. A. Kwong, "Protease circuits for processing biological information," *Nature Communications*, vol. 11, no. 1, pp. 1–12, Oct. 2020.

[180] T. Huynh, B. Sun, L. Li, K. P. Nichols, J. L. Koyner, and R. F. Ismagilov, "Chemical Analog-to-Digital Signal Conversion Based on Robust Threshold Chemistry and Its Evaluation in the Context of Microfluidics-Based Quantitative Assays," *Journal of the American Chemical Society*, vol. 135, no. 39, pp. 14 775–14 783, Sep. 2013.

[181] M. Moškon, Ž. Pušnik, L. Stanovnik, N. Zimic, and M. Mraz, "A computational design of a programmable biological processor," *Biosystems*, vol. 221, pp. 1–12, Nov. 2022.

[182] F.-L. A. Lau, R. Wendt, and S. Fischer, "Efficient in-message computation of prevalent mathematical operations in DNA-based nanonetworks," *Elsevier Nano Communication Networks*, vol. 28, pp. 1–17, Jun. 2021.

[183] M. S. Diab and E. Rodriguez-Villegas, "Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care Applications: A Review," *IEEE Access*, vol. 10, pp. 98 450–98 474, 2022.

[184] A. Faraone and R. Delgado-Gonzalo, "Convolutional-Recurrent Neural Networks on Low-Power Wearable Platforms for Cardiac Arrhythmia Detection," in *2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS 2020)*, Genova, Italy: IEEE, Aug. 2020.

[185] *nRF52832 Product Specification*, en. [Online]. Available: docs.arduino.cc/resources/datasheets/nRF52832_PS_v1.1.pdf.

[186] *96 Well Cell Culture Microplate CELLCOAT*, en. [Online]. Available: shop.gbo.com/en/germany/files/27485048/655936.pdf.

[187] 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.

[188] M. Nauta et al., "From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, Jul. 2023.

[189] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA: ACM, Aug. 2016, pp. 1135–1144.

[190] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *31st International Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA: Curran Associates Inc., Dec. 2017, pp. 4768–4777.

[191] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers," in *Artificial Neural Networks and Machine Learning*, A. E.P. Villa, P. Masulli, and A. J. Pons Rivero, Eds., Barcelona, Spain: Springer International Publishing, 2016, pp. 63–71.

[192] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *34th International Conference on Machine Learning (ICML 2017)*, Sydney, Australia: JMLR.org, Aug. 2017, pp. 3145–3153.

[193] T. N. Mundhenk, B. Y. Chen, and G. Friedland, "Efficient Saliency Maps for Explainable AI," arXiv, cs.CV 1911.11293, Mar. 2019.

[194] A. Wachter-Zeh, B. Mittelstadt, and C. Russell, "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR," arXiv, cs.AI, Mar. 2017, pp. 1–52.

[195] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[196] X. Li, "Explainability of NN-based Detectors in MIMO Molecular Channels," Master's Thesis, School of Electrical Engineering and Computer Science, Berlin, Germany, Jul. 2023.

[197] J. Torres Gómez, P. Hofmann, F. H. P. Fitzek, and F. Dressler, "Explainability of Neural Networks for Symbol Detection in Molecular Communication Channels," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 3, pp. 323–328, Sep. 2023.

[198] Y. Huang, M. Luo, X. Huang, M. Wen, and C.-B. Chae, "Demystifying Molecular Data-driven Detection with Explainable Artificial Intelligence," *IEEE Wireless Communications Letters*, 2025.

[199] O. T. Basaran, J. Torres Gómez, and F. Dressler, "XAI-Enhanced Bilateral Molecular Communication: Revealing Cancer Microenvironment Dynamics via Extracellular Tumor Vesicles," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2025)*, Barcelona, Spain: IEEE, May 2025, pp. 1–6.

[200] H. C. Berg, *Random Walks in Biology*. Princeton University Press, 1993, p. 152.

[201] N. Tuccitto, *Dataset of "Experimental Implementation of Molecule Shift Keying for Enhanced Molecular Communication"*, IEEE Dataport, 2023. [Online]. Available: doi.org/10.21227/krbw-ge80.

[202] T. Nunzio, *Dataset of "Experimental Implementation of Molecule Shift Keying for Enhanced Molecular Communication"*, Zenodo, Dec. 2023. [Online]. Available: doi.org/10.5281/zenodo.10390855.

[203] P. Hofmann, J. Torres Gómez, F. H. Frank H.P., and F. Dressler, *Dataset for Macroscale Molecular Communication Testbed*, IEEE Dataport, 2023. [Online]. Available: doi.org/10.21227/ytkm-xp81.

[204] P. Hofmann, P. Zhou, C. Lee, M. Reisslein, F. H. P. Fitzek, and C.-B. Chae, "OpenFOAM Simulation of Microfluidic Molecular Communications: Method and Experimental Validation," *IEEE Access*, vol. 12, pp. 109 494–109 512, 2024.

[205] P. Hofmann, P. Zhou, C. Lee, M. Reisslein, F. H. Fitzek, and C.-B. Chae, *Dataset for the Simulation of Microfluidic Molecular Communication using OpenFOAM*, IEEE Dataport, 2023. [Online]. Available: doi.org/10.21227/b71c-4286.

[206] P. Zhou, *DMPPIC*, 2024. [Online]. Available: github.com/zhoupengjie/DMPPIC.

[207] A. Wietfeld et al., "Advanced Plaque Modeling for Atherosclerosis Detection Using Molecular Communication," arXiv, cs.ET, Nov. 2024, pp. 1–6.

[208] P. Hofmann et al., "A Molecular Communication Perspective on Detecting Arterial Plaque Formation," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 3, pp. 458–463, Sep. 2024.

[209] P. Hofmann et al., *Dataset for Advanced Plaque Modeling for Atherosclerosis Detection using Molecular Communication*, IEEE Dataport, 2024. [Online]. Available: doi.org/10.21227/mn8d-0h55.

[210] F. Gulec, *CFD Simulation Dataset for Airborne Pathogen Transmission in Turbulent Channels*, Zenodo, 2024. [Online]. Available: doi.org/10.5281/zenodo.13793238.

[211] F. Güleç, F. Dressler, and A. W. Eckford, "A Computational Approach for the Characterization of Airborne Pathogen Transmission in Turbulent Molecular Communication Channels," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 2, pp. 124–134, Jun. 2023.

[212] F. Gulec, *Distance-Estimation-in-Molecular-Communication*, 2020. [Online]. Available: github.com/fatihguelec/Distance-Estimation-in-Molecular-Communication.

[213] F. Güleç and B. Atakan, "Fluid dynamics-based distance estimation algorithm for macroscale molecular communication," *Elsevier Nano Communication Networks*, vol. 28, pp. 1–9, Jun. 2021.

[214] W. Gao, *Dataset in Support of the Southampton Doctoral Thesis "Type-Spread and Multiple-Access Molecular Communications"*, 2023. [Online]. Available: doi.org/10.5258/SOTON/D2545.

[215] W. Gao, T. Mak, and L.-L. Yang, "Molecular Type Spread Molecular Shift Keying for Multiple-Access Diffusive Molecular Communications," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 7, no. 1, pp. 51–63, Mar. 2021.

[216] W. Gao, "Type-Spread and Multiple-Access Molecular Communications," PhD Thesis, School of Electronics and Computer Science, Southampton, United Kingdom, Feb. 2023.

[217] A. Das, B. Runwal, O. T. Baydas, O. Cetinkaya, and O. B. Akan, *Received Signal Modeling and BER Analysis for Molecular SISO Communications*, Zenodo, 2022. [Online]. Available: doi.org/10.5281/zenodo.7036057.

[218] A. Das, B. Runwal, O. T. Baydas, O. Cetinkaya, and O. B. Akan, "Received signal modeling and BER analysis for molecular SISO communications," in *9th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2022)*, Barcelona, Spain, Oct. 2022, pp. 1–6.

[219] M. Hamidović, S. Angerbauer, D. Bi, Y. Deng, T. Tugcu, and W. Haselmayr, "Microfluidic Systems for Molecular Communications: A Review From Theory to Practice," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 1, pp. 147–163, Mar. 2024.

[220] K. Gökarslan and E. Çağırıcı, *MoleCom-Gpu*, 2017. [Online]. Available: github.com/MoleCom-Gpu/MoleCom-Gpu.

[221] Y. Jian et al., "nanoNS3: A network simulator for bacterial nanonetworks based on molecular communication," *Elsevier Nano Communication Networks*, vol. 12, pp. 1–11, Jun. 2017.

[222] G. Wei, P. Bogdan, and R. Marculescu, "Efficient Modeling and Simulation of Bacteria-Based Nanonetworks with BNSim," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 868–878, Dec. 2013.

[223] B. Drawert, S. Engblom, and A. Hellander, "URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries," *BMC Systems Biology*, vol. 6, no. 1, pp. 1–17, Jun. 2012.

[224] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, "Detailed Simulations of Cell Biology with Smoldyn 2.1," *PLoS Computational Biology*, vol. 6, no. 3, pp. 1–10, Mar. 2010.

[225] S. S. Andrews, "Smoldyn: particle-based simulation with rule-based modeling, improved molecular interaction and a library interface," *Bioinformatics*, vol. 33, no. 5, pp. 710–717, Dec. 2016.

[226] I. Llatser et al., "Exploring the Physical Channel of Diffusion-Based Molecular Communication by Simulation," in *IEEE Global Communications Conference (GLOBECOM 2011)*, Houston, TX: IEEE, Dec. 2011, pp. 1–5.

[227] I. Llatser, D. Demiray, A. Cabellos-Aparicio, D. T. Altilar, and E. Alarcón, "N3Sim: Simulation framework for diffusion-based molec-

ular communication nanonetworks," *Elsevier Simulation Modelling Practice and Theory*, vol. 42, pp. 210–222, Mar. 2014.

[228] A. Noel, K. C. Cheung, R. Schober, D. Makrakis, and A. Hafid, "Simulating with AcCoRD: Actor-based Communication via Reaction–Diffusion," *Elsevier Nano Communication Networks*, vol. 11, pp. 44–75, Mar. 2017.

[229] X. Qian, M. Di Renzo, and A. Eckford, *Molecular-Communication*, 2020. [Online]. Available: github . com / MuskanM1 / Molecular - Communication.

[230] P. Sangani, *Research-in-Molecular-Communication*, 2021. [Online]. Available: github . com / Priyank31 / Research - in - Molecular - Communication.

[231] D. Patel, *Molecular-Communication*, 2020. [Online]. Available: github.com/devshree07/Molecular-Communications.

[232] Y. Shastri, *Molecular-Communication—Model-based-and-Data-Driven-Receiver-Design*, 2020. [Online]. Available: github . com / Yesha19 / Molecular - Communication --- Model - based - and - Data - Driven-Receiver-Design.

[233] H. Birkan, *ANN for Diffusion Channel with Reflecting Spherical to Absorbing Spherical*, MATLAB Central File Exchange, 2017. [Online]. Available: mathworks.com/matlabcentral/fileexchange/61382-ann-for-diffusion-channel-with-reflecting-spherical-to-absorbing-spherical.

[234] F. Gulec, *Signal-Reconstruction-in-Diffusion-based-Molecular-Communication*, 2018. [Online]. Available: github . com / fatihguelec/Signal - Reconstruction - in - Diffusion - based - Molecular - Communication.

[235] B. Atakan and F. Güleç, "Signal reconstruction in diffusion-based molecular communication," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 12, pp. 1–14, Dec. 2019.

[236] F. Gulec, *Mobile-Human-Ad-Hoc-Networks*, 2022. [Online]. Available: github.com/fatihguelec/Mobile-Human-Ad-Hoc-Networks.

[237] F. Güleç, B. Atakan, and F. Dressler, "Mobile Human Ad Hoc Networks: A Communication Engineering Viewpoint on Interhuman Airborne Pathogen Transmission," *Elsevier Nano Communication Networks*, vol. 32-33, pp. 1–11, Jun. 2022.

[238] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, May 2020.

[239] F. Gulec and A. Eckford, "Stochastic Modeling of Biofilm Formation with Bacterial Quorum Sensing," in *IEEE International Conference on Communications (ICC 2023)*, Rome, Italy: IEEE, May 2023, pp. 4470–4475.

[240] F. Gulec and A. Eckford, "A Stochastic Biofilm Disruption Model Based on Quorum Sensing Mimickers," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 3, pp. 346–350, Sep. 2023.

[241] F. Gulec, *Stochastic-Modeling-of-Biofilm-Formation-with-Bacterial-Quorum-Sensing*, 2022. [Online]. Available: github.com/fatihguelec/Stochastic-Modeling-of-Biofilm-Formation-with-Bacterial-Quorum-Sensing.

[242] F. Gulec, *Stochastic-Biofilm-Disruption-Model-Based-on-Quorum-Sensing-Mimickers*, 2022. [Online]. Available: github.com/fatihguelec/Stochastic-Biofilm-Disruption-Model-Based-on-Quorum-Sensing-Mimickers.

[243] Z. Zhang, *Informer-based*, 2024. [Online]. Available: github.com/Zhichao-Zhang-Zjut/Informer-based.

[244] D. Scazzoli, F. Vakilipoor, and M. Magarini, "Molecular communication data augmentation and deep learning based detection," *Elsevier Nano Communication Networks*, vol. 40, pp. 1–12, Jul. 2024.

[245] L. Grebenstein et al., "Biological optical-to-chemical signal conversion interface: a small-scale modulator for molecular communications," in *5th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2018)*, Reykjavík, Iceland: ACM, Sep. 2018, pp. 1–6.

[246] P. Hofmann, J. A. Cabrera, R. Bassoli, and F. H. Fitzek, *Dataset for Analog Network Coding in Molecular Communications: A Practical Implementation*, IEEE Dataport, 2023. [Online]. Available: doi.org/10.21227/57z0-9q64.

[247] M. Bartunik, *Channel Parameter Studies with a Biocompatible Testbed for Molecular Communication*, IEEE Dataport, 2023. [Online]. Available: doi.org/10.21227/g15d-kz12.

[248] F. Calì, G. Li-Destri, and N. Tuccitto, *The Data Related to Interfacial Shift Keying Allows a High Information Rate in Molecular Communication*, IEEE Dataport, Oct. 2022. [Online]. Available: doi.org/10.21227/mj9p-pt58.

[249] F. Calì, G. Li-Destri, and N. Tuccitto, *The Data Related to Interfacial Shift Keying Allows a High Information Rate in Molecular Communication*, version 1.0.0, Zenodo, 2022. [Online]. Available: doi.org/10.5281/zenodo.7244181.

[250] M. Abbaszadeh, I. Atthanayake, P. J. Thomas, and W. Guo, *Molecular Signal Tracking and Detection Methods in Fluid Dynamic Channels*, IEEE Dataport, 2019. [Online]. Available: doi.org/10.21227/ynet-ss80.

[251] M. Abbaszadeh, I. Atthanayake, P. J. Thomas, and W. Guo, *Molecular Signal Tracking and Detection Methods in Fluid Dynamic Channels (Method and Data)*, IEEE Dataport, Jan. 2020. [Online]. Available: doi.org/10.21227/eae4-kg81.

[252] L. Grebenstein et al., *A Molecular Communication Testbed Based on Proton Pumping Bacteria*, IEEE Dataport, 2019. [Online]. Available: doi.org/10.21227/3zj6-pm05.

[253] V. Walter, D. Bi, A. Salehi-Reyhani, and Y. Deng, *Real-Time Signal Processing via Chemical Reactions for a Microfluidic Molecular Communication System*, Zenodo, 2023. [Online]. Available: doi.org/10.5281/zenodo.8422465.

[254] M. Scherer et al., *Media Modulation Testbed: Code Package*, 2025. [Online]. Available: github.com/SyMoCADS/Media_Modulation_Testbed/.

[255] M. Scherer et al., *Closed-Loop Long-Term Experimental Molecular Communication System*, Zenodo, Feb. 2025. [Online]. Available: doi.org/10.5281/zenodo.13898880.

[256] F. Vakilipoor, A. Ettner-Sitter, R. Schober, S. Haerteis, and M. Schäfer, *Transient Indocyanine Green Distribution Measurement and Modelling in Chorioallantoic Membrane (CAM) Model*, Zenodo, Mar. 2025. [Online]. Available: doi.org/10.5281/zenodo.14626107.

[257] F. Calì, S. Barreca, G. Li-Destri, A. Torrisi, A. Licciardello, and N. Tuccitto, "Experimental Implementation of Molecule Shift Keying for Enhanced Molecular Communication," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 10, no. 1, pp. 175–184, Mar. 2024.

[258] P. Hofmann, J. A. Cabrera, R. Bassoli, and F. H. P. Fitzek, "Analog Network Coding in Molecular Communications: A Practical Implementation," in *IEEE Global Communications Conference (GLOBECOM 2023)*, Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 571–576.

[259] M. Bartunik, J. Teller, G. Fischer, and J. Kirchner, "Channel Parameter Studies of a Molecular Communication Testbed With Biocompatible Information Carriers: Methods and Data," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 4, pp. 489–498, Dec. 2023.

[260] A. Noel, *AcCoRD*, 2020. [Online]. Available: github.com/adamjgnoel/AcCoRD.

[261] L. Felicetti, M. Femminella, and G. Reali, "A simulation tool for nanoscale biological networks," *Elsevier Nano Communication Networks*, vol. 3, no. 1, pp. 2–18, Mar. 2012.

[262] L. Felicetti, M. Femminella, and G. Reali, "Simulation of molecular signaling in blood vessels: Software design and application to atherogenesis," *Elsevier Nano Communication Networks*, vol. 4, no. 3, pp. 98–119, Sep. 2013.

[263] L. A. Harris et al., "BioNetGen 2.2: advances in rule-based modeling," *Bioinformatics*, vol. 32, no. 21, pp. 3366–3368, Jul. 2016.

[264] *BioNetGen website*. [Online]. Available: https://bionetgen.org/.

[265] R. Geyer, C. Deter, and S. Fischer, "BVS-Vis: a web-based visualizer for BloodVoyagerS," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020)*, Virtual Conference: ACM, Sep. 2020.

[266] R. Wendt and S. Fischer, "MEHLISSA: A Medical Holistic Simulation Architecture for Nanonetworks in Humans," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020)*, Virtual Conference: ACM, Sep. 2020.

[267] R. Wendt, *blood-voyager-s*, 2020. [Online]. Available: github.com/RegineWendt/blood-voyager-s.

[268] R. Wendt, *BVS-Vis*, 2020. [Online]. Available: github . com / RegineWendt/BVS-Vis.

[269] R. Wendt, *MEHLISSA*, 2024. [Online]. Available: github . com / RegineWendt/MEHLISSA.

[270] L. Ebner et al., "BVS-Net: A Networking Tool for Studying THz-based Intra-body Communication Links," in *11th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2024), Poster Session*, Milan, Italy: ACM, Oct. 2024, pp. 132–133.

[271] R. Wendt, L. Ebner, and J. T. Gomez, *BVS_Net*, 2019. [Online]. Available: github.com/tkn-tub/BVS_Net.

[272] *BNSim website*. [Online]. Available: https://radum.ece.utexas.edu/bnsim-bacteria-network.

[273] J. Hattne, D. Fange, and J. Elf, "Stochastic reaction-diffusion simulation with MesoRD," *Bioinformatics*, vol. 21, no. 12, pp. 2923–2924, Apr. 2005.

[274] J. Hattne, D. Fange, and J. Elf, *MesoRD*, 2005. [Online]. Available: sourceforge.net/projects/mesord/.

[275] B. Morgan, *MolComSim*, 2015. [Online]. Available: github.com/calypsomatic/MolComSim.

[276] H. B. Yilmaz and C.-B. Chae, "Simulation study of molecular communication systems with an absorbing receiver: Modulation and ISI mitigation techniques," *Elsevier Simulation Modelling Practice and Theory*, vol. 49, pp. 136–150, Dec. 2014.

[277] H. Birkan, *MolecUlar CommunicatIoN (MUCIN) Simulator*, 2024. [Online]. Available: mathworks.com/matlabcentral/fileexchange/46066-molecular-communication-mucin-simulator.

[278] T. Saiki and T. Nakano, "Design and Implementation of a Multicellular Molecular Communication Simulator," in *Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS and ISIS 2022)*, Ise, Japan: IEEE, Nov. 2022, pp. 1–5.

[279] T. Saiki, S. Imanaka, S. Kobayashi, and T. Nakano, "A General-Purpose Simulation Platform for Multicellular Molecular Communication Systems," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 11, no. 2, pp. 152–165, Jun. 2025.

[280] T. Saiki and S. Imanaka, *Multicellular Molecular Communication System Simulator*, 2025. [Online]. Available: github.com/ImanakaShohei/MulticellularMolecularCommunicationSystemSimulator.

[281] *Download link*. [Online]. Available: http://gnan.ece.gatech.edu/ns-allinone-3.24.zip.

[282] *N3Sim*. [Online]. Available: https://n3cat.upc.edu/n3sim-simulation-framework - for - diffusion - based - molecular - communication - nanonetworks/.

[283] N. A. Turgut, B. A. Bilgin, and O. B. Akan, "N^4 Sim: The First Nervous NaNoNetwork Simulator With Synaptic Molecular Communications," *IEEE Transactions on NanoBioscience*, vol. 21, no. 4, pp. 468–481, Oct. 2022.

[284] N. A. Turgut, *N4Sim*, 2020. [Online]. Available: github.com/nafiturgut/N4Sim.

[285] M. Ander et al., "SmartCell, a framework to simulate cellular processes that combines stochastic approximation with diffusion and localisation: analysis of simple networks," *IEEE Proceedings – Systems Biology*, vol. 1, no. 1, pp. 129–138, Jun. 2004.

[286] *Smoldyn*. [Online]. Available: https://www.smoldyn.org/.

[287] P. Bauer, S. Engblom, A. Senek, and D. Wilson, *URDME (Version 1.4)*, 2024. [Online]. Available: github.com/URDME/urdme.

[288] A. Grimmer, M. Hamidović, W. Haselmayr, and R. Wille, "Advanced Simulation of Droplet Microfluidics," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 15, no. 3, pp. 1–16, Apr. 2019.

[289] G. Fink, F. Costamoling, and R. Wille, "MMFT Droplet Simulator: Efficient Simulation of Droplet-based Microfluidic Devices," *Software Impacts*, vol. 14, p. 100 440, Dec. 2022.

[290] A. Grimmer, M. Hamidović, W. Haselmayr, and R. Wille, *SIMPAC-2022-234*, 2022. [Online]. Available: github.com/SoftwareImpacts/SIMPAC-2022-234.

[291] J. P. Drees et al., "Efficient Simulation of Macroscopic Molecular Communication: The Pogona Simulator," in *7th ACM International Conference on Nanoscale Computing and Communication (NANOCOM 2020)*, Virtual Conference: ACM, Sep. 2020.

[292] L. Stratmann, J. P. Drees, F. Bronner, and F. Dressler, "Using Vector Fields for Efficient Simulation of Macroscopic Molecular Communication," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications, Special Section - Advances in Molecular Communication*, vol. 7, no. 2, pp. 73–77, Jun. 2021.

[293] J. P. Drees et al., *Pogona*, 2020. [Online]. Available: github.com/tkn-tub/pogona.

[294] F. Calì, G. Li-Destri, and N. Tuccitto, "Interfacial Shift Keying Allows a High Information Rate in Molecular Communication: Methods and Data," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 9, no. 3, pp. 300–307, Sep. 2023.

[295] M. Abbaszadeh, I. Atthanayake, P. J. Thomas, and W. Guo, "Molecular Signal Tracking and Detection Methods in Fluid Dynamic Channels," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 6, no. 2, pp. 151–159, Nov. 2020.

[296] L. Grebenstein et al., "A Molecular Communication Testbed Based on Proton Pumping Bacteria: Methods and Data," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 5, no. 1, pp. 56–62, Oct. 2019.

[297] V. Walter, D. Bi, and Y. Deng, *MolCommUI*, 2023. [Online]. Available: github.com/kcl-yansha/MolCommUI.

[298] L. Brand et al., "Closed Loop Molecular Communication Testbed: Setup, Interference Analysis, and Experimental Results," in *IEEE International Conference on Communications (ICC 2024)*, Denver, CO: IEEE, Jun. 2024, pp. 4805–4811.

[299] L. Brand et al., "Switchable Signaling Molecules for Media Modulation: Fundamentals, Applications, and Research Directions," *IEEE Communications Magazine*, vol. 62, no. 5, pp. 112–118, May 2024.

[300] M. Scherer et al., "Closed-Loop Long-Term Experimental Molecular Communication System," arXiv, cs.ET, Feb. 2025, pp. 1–30.

[301] F. Vakilipoor et al., "The CAM Model: An in Vivo Testbed for Molecular Communication Systems," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, Aug. 2025, early access.

[302] A. Das, B. Runwal, O. Cetinkaya, and O. B. Akan, *Channel Estimation and Performance Analysis of SISO Molecular Communications - Version v2*, Zenodo, 2021. [Online]. Available: doi.org/10.5281/zenodo.5701559.

[303] M. Thakkar, *molecular_communication*, Kaggle, 2020. [Online]. Available: kaggle.com/datasets/mithilesh16/molecular-communication.

[304] X. Lu, *MCFormer*, 2023. [Online]. Available: github.com/Xiwen-Lu/MCFormer.

[305] Y. Zhao, J. Chen, and S. Oymak, "On the Role of Dataset Quality and Heterogeneity in Model Confidence," arXiv, cs.LG, Feb. 2020. [Online]. Available: https://arxiv.org/abs/2002.09831.

[306] J. M. Johnson and T. M. Khoshgoftaar, "Survey on Deep Learning with Class Imbalance," *Journal of Big Data*, vol. 6, no. 1, Mar. 2019.

[307] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A survey on dataset quality in machine learning," *Information and Software Technology*, vol. 162, p. 107 268, Oct. 2023.

[308] A. Wietfeld et al., "Advanced Plaque Modeling for Atherosclerosis Detection Using Molecular Communication," in *IEEE International Conference on Communications (ICC 2025)*, Montréal, Canada: IEEE, Jun. 2025, pp. 6056–6062.

[309] F. Calì et al., "Fluorescent nanoparticles for reliable communication among implantable medical devices," *Carbon*, vol. 190, pp. 262–275, Apr. 2022.

[310] F. Gulec, *CFD-Approach-for-the-Characterization-of-Airborne-Transmission-in-Turbulent-MC-Channels*, 2024. [Online]. Available: github.com/fatihguelec/CFD-Approach-for-the-Characterization-of-Airborne-Transmission-in-Turbulent-MC-Channels.

[311] A. Das, B. Runwal, O. Cetinkaya, and O. B. Akan, *Channel Estimation and Performance Analysis of SISO Molecular Communications - Version v1*, Zenodo, 2021. [Online]. Available: doi.org/10.5281/zenodo.5701433.

[312] L. Ewart et al., "Performance assessment and economic analysis of a human Liver-Chip for predictive toxicology," *Nature Communications Medicine*, vol. 2, no. 1, pp. 1–16, Dec. 2022.

[313] G. Pascual, F. Lemic, C. Delgado, and X. Costa-Pérez, "Analytical Modelling of Raw Data for Flow-Guided In-body Nanoscale Localization," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden: IEEE, May 2024, pp. 428–433.

[314] L. Wei, Z. Ma, C. Yang, and Q. Yao, "Advances in the Neural Network Quantization: A Comprehensive Review," *Applied Sciences*, vol. 14, no. 17, Aug. 2024, Art. no. 7445.

[315] Y. He and L. Xiao, "Structured Pruning for Deep Convolutional Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2900–2919, Nov. 2024.

[316] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," *IEEE Access*, vol. 11, pp. 96 892–96 922, 2023.