Segmentation of single-cell impedance signals using deep learning: a multi-dataset study

Marta Righetto, Riccardo Reale, Adele De Ninno, Cristian Brandi, Daniel C. Spencer, Xueping Zou, Javad Jarmoshti, Nathan S. Swami, Hywel Morgan, Paolo Bisegna, Federica Caselli *

***Abstract*— *Objective*: Microfluidic impedance cytometry (MIC) is a high-throughput, label-free technology for single-cell analysis, with applications in cell classification and non-invasive monitoring. However, developing a universal signal-processing approach for MIC is challenging, because signal characteristics strongly depend on the experimental setup. This study investigates the integration of MIC with deep learning (DL) to enable effective processing of raw impedance signals from various MIC systems. Specifically, the research focuses on signal segmentation (i.e., event detection), which represents the first step of the processing pipeline and therefore influences the entire workflow. *Methods*: Impedance data from multiple experimental setups were collected. They are characterized by raw traces with diverse attributes and event signals exhibiting distinct temporal shapes, thus forming a rich and comprehensive database. Several DL models were implemented and compared, including recurrent, convolutional, and encoder–decoder neural networks. *Results*: While all models demonstrated good segmentation performance, the encoder–decoder network outperformed the others, achieving a sensitivity and positive predictive value of 91.6% and 91.8%, respectively. Moreover, the network remained robust when, after training, validation, and testing, it was further evaluated on additional previously unseen data. *Conclusion*: We developed a framework for signal segmentation in MIC that addresses the challenge of cross-setup generalizability. *Significance*: By enabling efficient, high-speed processing, the integration of MIC and DL lays the foundation for next-generation single-cell workflows with applications in diagnostics, drug discovery, and environmental monitoring.**

***Index Terms*— Microfluidic impedance cytometry, single-cell analysis, time series, segmentation, neural networks, deep learning, cross-setup generalizability.**

I. Introduction

MICROFLUIDIC impedance cytometry (MIC) is a non-invasive technique that characterizes single cells based on their interaction with a multifrequency electric field [1]. In a typical impedance cytometry setup, cells suspended in a conductive buffer flow one-by-one through a microchannel with integrated microelectrodes (referred to as a microfluidic chip). An AC voltage is applied to a subset of electrodes and the current from the remaining electrodes is measured. Most frequently, a differential measurement scheme is adopted [2].

The electrical current change measured from a cell depends on the stimulation frequency [3]. For cells in saline, at low frequencies (<1 MHz), the response mainly reflects cell size due to the insulating effect of the membrane. At intermediate frequencies (1–10 MHz), membrane polarization dominates and membrane capacitance can be determined. At higher frequencies (>10 MHz), the signal reflects the dielectric properties of the cytoplasm and, eventually, the nucleus. Furthermore, tailored chip designs and wiring schemes enable measurement of additional cell properties, including cell shape [4], [5] and deformability [6], [7], [8], [9], or cell-motion properties [10], [11]. MIC thus enables single-cell biophysical analysis and stratification of cellular heterogeneity, with versatility in life sciences, drug discovery, diagnostics, and environmental research (see e.g., the reviews [1], [12], [13] and the references therein).

To extract the electrical features of each flowing cell from the measured data streams, the first processing step is signal segmentation, i.e., identification of the signal segments corresponding to cells passing through the sensing region, also known as event detection. Signal segmentation in MIC is usually performed via thresholding and peak-finding, which can be implemented on the conditioned signal [14], in the wavelet-domain [15], or after correlation with appropriate templates/signals [16], [17], [18]. However, these approaches are generally tailored to a specific experimental setup, rely on critical parameters (e.g., minimum peak height, template features) that require fine tuning, and often lack a quantitative analysis of their performance [19]. Moreover, they are typically implemented offline, since their computational cost may be incompatible with real-time processing.

Recently, the application of machine learning (ML) tools to impedance cytometry data has attracted increasing attention [20]. ML techniques can be applied across two distinct domains: the feature space and the signal space. The feature-space approach is the most widely adopted methodology and involves applying ML models to electrical features extracted using traditional algorithms (e.g., [21] [22], [23], [24], [7], [25], [26]). Conversely, the signal-space approach operates directly on raw data streams, potentially capturing richer information and simplifying the processing pipeline, although

This research was funded by the European Union – Next Generation EU, Mission 4, Component 1 – CUP E53D23002530006 (grant 2022245PTX, MIC_AIM project, PRIN 2022 programme) and by the University of Rome Tor Vergata (grant NN4RTA_RSA_2024).

M. Righetto, R. Reale, C. Brandi, P. Bisegna, and F. Caselli are with the Department of Civil Engineering and Computer Science, University of Rome Tor Vergata 00133, Rome, Italy (e-mail: caselli@ing.uniroma2.it).

A. De Ninno is with the Institute for Photonics and Nanotechnology, Italian National Research Council, 00133, Rome, Italy.

D. C. Spencer, X. Zou and H. Morgan are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK.

J. Jarmoshti and N. S. Swami are with the Electrical & Computer Engineering Department, University of Virginia, Charlottesville, VA, 22904, USA.

it remains less explored. A few deep learning (DL) models have been proposed. For example, Honrado et al. [27] developed a recurrent neural network (RNN) to perform real-time feature extraction from single-cell MIC signals, demonstrating accurate characterization of size, velocity, and cross-sectional position of red blood cells (RBCs) and yeasts. Caselli et al. [28] combined RNNs and convolutional neural networks (CNNs) to decipher MIC signals in two challenging scenarios: fast dielectric spectroscopy of RBCs and resolution of single-cell signals that are hidden in the signals of coincident cells. Neural networks designed for the fast processing of challenging signal waveforms have also been introduced in closely related application fields, including distributed Coulter sensors [29], [30] and magnetic flow cytometry [31]. While those studies demonstrate promising results, their networks were trained on a specific chip design and signal configuration, limiting their generalizability. A study assessing the applicability of neural networks to process MIC data streams acquired across multiple experimental setups is still missing.

The present work aims to develop a robust and universal neural network for the segmentation of MIC raw signals. The neural network should be able to perform event detection regardless of the specific chip architecture or the details of the event-signal temporal shape, as long as a differential measurement scheme is used (i.e., event signals are bipolar). To achieve this, we collected MIC data streams from several experimental setups associated with previously published papers. As detailed in Section II-A, we built a rich and comprehensive database that includes datasets acquired with coplanar- [2], facing- [32], or liquid- [33] electrode chip designs, using varied wiring schemes. Accordingly, the event-signal shapes may have a symmetric or asymmetric bipolar profile, with or without secondary peaks (i.e., Gaussian or Mexican hat pulses), and possibly occur in pairs (i.e., two close bipolar profiles for each flowing cell). The datasets cover a broad range of samples: polystyrene beads, bacteria, RBCs, pollen grains, human cell lines, cell xenografts, and cell-loaded nanovials. In each case, microchannel dimensions were consistent with particle sizes, and the flow rate Φ was dependent on the application (e.g., lower for applications involving dielectrophoretic manipulation in addition to sensing). The datasets were acquired using distinct configurations of the electronic acquisition system (in particular, stimulation frequencies and sampling frequency f_s) and are characterized by heterogeneous particle throughput (i.e., measured events per second) and varying levels of signal-to-noise ratio (SNR). To the best of our knowledge, this is the first study to validate DL segmentation models across a heterogeneous spectrum of MIC hardware (chip layouts, wiring schemes, measurement settings), sample types, and corresponding event-signal morphologies.

As detailed in Section II-C, we investigated the following network architectures: three RNNs with long short-term memory (LSTM), bidirectional LSTM (biLSTM), or gated recurrent units (GRU); a CNN with a simple sequential structure; a temporal convolutional network (TCN) [34]; and an encoder–decoder network (EDN) [35]. All networks receive as input fixed-length blocks of impedance data (referred to as chunks) and produce as output a binary segmentation mask, where '1' indicates signal segments associated with the passage of a particle through the sensing zone and '0' marks idle intervals between particles. Network performance was evaluated using overall 0/1 classification accuracy (A), as well as sensitivity (S) and positive predictive value (PPV) in the event-detection task.

The results of the study are described in Section III, while Section IV discusses strengths, limitations, and future perspectives. Finally, conclusions are drawn in Section V.

II. Methodology

A. Database

The datasets analysed in this study are sourced from previous work from the authors laboratories as follows: [36], [37], [22], [38], [39], [40], [7], [8], and [41]. While we refer the reader to the original papers for thorough analysis, Table I summarizes the most relevant features of each dataset ($01-09$). Figure 1 illustrates the various chip layouts and wiring schemes that were used in those works, together with the shapes of the associated event signals. The correspondence between each dataset and its wiring scheme is given in Table I – sixth column. While idealized event signals are considered in Fig. 1, an example of raw signals from dataset 01 is shown in Fig. 2. Analogous pictures for all the datasets are provided in Fig. S1 of the Supplementary Material.

As is customary in MIC, most of the datasets are acquired at two simultaneous stimulation frequencies: a "low" frequency below 1 MHz (f_L) and a "high" frequency of several MHz (f_H). This gives four signal channels (i.e., real and imaginary parts of the signal for each frequency). Accordingly, we designed the networks for a four-channel input. For dataset 01, where eight frequencies were originally measured, we picked 0.23 MHz and 10 MHz as low and high frequencies; for datasets 04 and 06, where only one frequency was available, we duplicated the data. Furthermore, for each dataset we generated a companion dataset where the real and imaginary parts of the high frequency channel were set to 0. In this way, the developed models will also be applicable to single-frequency MIC measurements. Data stream duration is 70 s for each dataset, except for dataset 01 (because only 44 s were originally acquired) and dataset 06 (for which we considered a longer duration of 30 min, due to the very low flow rate).

It is noticed that the datasets collected in this study are relevant to MIC setups with a limited number of coincidences. Accordingly, in the original works no special strategy was implemented to handle coinciding events. Likewise, the DL models developed here do not address coincidence arbitration. To resolve coincidences, dedicated algorithms have been proposed in the literature (e.g., Bayesian approach [42], sparse deconvolution [43], sparse constrained optimization [44], data-driven threshold selection [45], or custom neural network architectures [28], [29], [46]), frequently coupled with tailored chip layouts (e.g., multi-electrode layouts or multiple

measuring zones).

Table I. Dataset summary table. The collection of datasets 01−08 constitutes the database used for training, validation, and testing; dataset 09 was reserved for further testing.

ID	SubID	Ref.	Original application	Sample type	Layout as in	W (μm) × H (μm)	Acquisition settings: Φ (μL/min), f_s (kHz), f_L (MHz) and f_H (MHz)	Attributes: λ (events/s), δ (sample), SNR
01	A	[36]	Dielectric spectroscopy	RBCs	Fig. 1(a)	40 × 30	Φ = 40, f_s = 56, f_L = 0.23 and f_H = 10	λ = 222, δ = 12.95, SNR = 31.6
02	A	[37]	Viability assessment	U937 lymphoma cells	Fig. 1(b)	40 × 21	Φ = 10, f_s = 115, f_L = 0.5 and f_H = 10	λ = 325, δ = 14.6, SNR = 6.0
03	A	[22]	Pollen grain classification	Pollen grains	Fig. 1(c)	120 × 63	Φ = 90, f_s = 115, f_L = 0.5 and f_H = 10	λ = 56.8, δ = 83.4, SNR = 13.3
04	A	[38]	Innovative wiring scheme	Polystyrene beads	Fig. 1(c)	40 × 21	Φ = 10, f_s = 115, f_L = 0.5 and f_H = N/A	λ = 121, δ = 28.7, SNR = 17.7
05	A	[39]	MIC-nanovials compatibility	Nanovials with MSCs	Fig. 1(b)	120 × 60	Φ = 50, f_s = 115, f_L = 0.5 and f_H = 20	λ = 27.4, δ = 84.5, SNR = 196.0
06	A−D	[40]	DEP monitoring	Polystyrene beads	Fig. 1(c)	40 × 21	Φ = 0.02 (A, B) or 0.04 (C, D), f_s = 0.9, f_L = 0.5 and f_H = N/A	λ = 0.28 (A, B) / 0.86 (C, D), δ = 115 (A, B) / 46.9 (C, D), SNR = 5.7 (A, B) / 16.5 (C, D)
07	A, B	[7]	Deformability assessment	Pancreatic cancer cells (xenografts)	Fig. 1(d) S_1 (A) / S_2 (B)	50−20 × 30	Φ = 12, f_s = 115, f_L = 0.5 and f_H = 18	λ = 163.5, δ = 30.1 (A) / 11.9 (B), SNR = 68.6 (A) / 233.7 (B)
08	A	[8]	Deformability assessment	HL60 leukemia cells	Fig. 1(e)	40 × 30	Φ = 5, f_s = 230, f_L = 0.5 and f_H = 5	λ = 145.8, δ = 124.9, SNR = 135.6
09	A, B	[41]	Toxicity testing	RBCs (A) / RBCs and *Bacillus megaterium* (B)	Fig. 1(b)	40 × 20	Φ = 10, f_s = 115, f_L = 0.5 and f_H = 20	λ = 60.4 (A) / 162.9 (B), δ = 16.7 (A) / 15.0 (B), SNR = 183.3 (A) / 37.4 (B)

(ID and SubID, dataset identifier and sub-identifier; W, microchannel width; H, microchannel height; Φ, flow rate; f_s, sampling frequency; f_L and f_H, low and high stimulation frequency, respectively; λ, event throughput; δ, median peak-to-peak distance; SNR, signal-to-noise ratio of real signal at f_L, computed as the median value of event amplitudes divided by noise standard deviation; MSCs, mesenchymal stromal cells; most samples also included polystyrene beads for calibration).

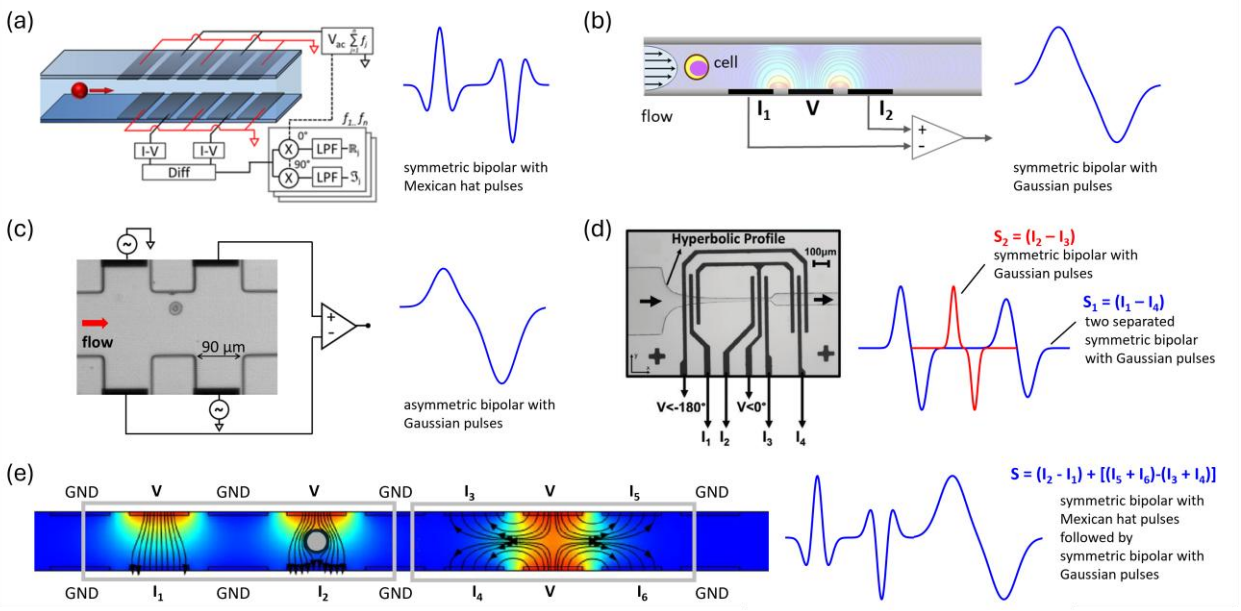

Fig. 1. Reference chip layouts and wiring schemes, along with the associated idealized event signals. Event-signal features (e.g., symmetric/asymmetric bipolar, Mexican hat/Gaussian pulses) are described within the panels. (a) Facing electrode design, five-electrode pair arrangement. (b) Coplanar electrode design, standard three-electrode differential measurement. (c) Liquid electrode design, differential wiring scheme with diagonally opposite stimulating electrodes. (d) Coplanar electrode design, three-electrode differential measurement repeated across three measurement zones and adjusted to have only two signal channels. (e) Facing electrode design, five-electrode pair arrangement (as in panel (a)) followed by a second measurement zone equivalent to that in panel (b) replicated top and bottom. Images were adapted with permission from (a) Ref. [36], copyright 2020 American Chemical Society, (b) Ref. [37], copyright 2019 Elsevier B.V., (c) Ref. [22], copyright 2021 IEEE, (d) Ref. [7], copyright 2024 The Author(s), and (e) Ref. [56], copyright The Author.

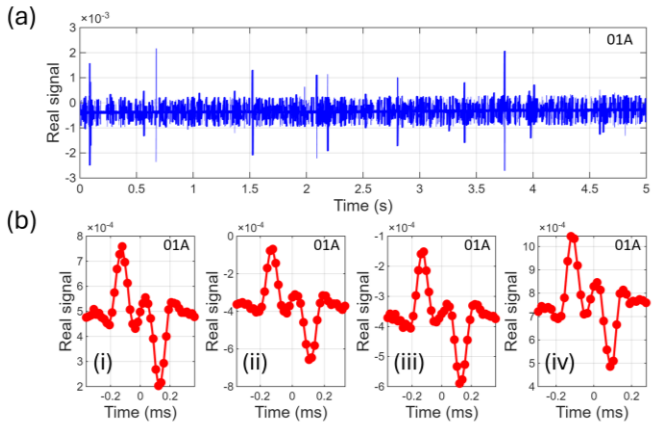

Fig. 2. (a) Exemplary portion (5 seconds) of a raw data stream (i.e., signal real part at low frequency, from dataset 01). (b) Four randomly chosen single-cell event signals (from dataset 01).

B. *Annotation and preprocessing*

For each dataset, the target segmentation mask was obtained by fine-tuning a reference algorithm based on preconditioning, correlation and peak detection, and template fitting. The algorithm is briefly described in Section S2 of the Supplementary Material. Several attributes were identified for each dataset, including (cf. Table I): throughput λ (i.e., number of events per second), median value of the event peak-to-peak distance (δ), and SNR.

The signal channels of each dataset were resampled to achieve a median peak-to-peak distance δ_{ref} of 16 samples. For each dataset, the resampling factor was computed as the rational fraction approximation of δ_{ref}/δ (with 10% tolerance). The MATLAB function `resample.m` was used, which applies an FIR antialiasing low-pass filter to the input sequence and compensates for the delay introduced by the filter. The signals were then split into chunks of length $L = 1000$ samples. This value was chosen based on the following considerations: L should be at least one order of magnitude larger than δ_{ref} to avoid excessive event truncation; L should be on the order of the average number of samples between two events (i.e., $f_s(\delta_{ref}/\delta)/\lambda$), so that on average each chunk contains a few events. In addition, at least 5% of the chunks should contain an event to properly tune the signal scale. For each chunk, a linear baseline was subtracted from each channel (estimated from the average values of the left and right half of the channel chunk), and the largest signal range across the four channels was taken as the chunk range. The dataset scale was defined as half of the 95th percentile of all chunk ranges, and each dataset was normalized by dividing its values by its scale. Six representative examples of normalized chunks are shown in Fig. 3 (event-rich chunks were selected) along with each target segmentation mask.

Overall, the database included 74,730 chunks from datasets 01−08. Of these, 85% was allocated to training (70%) and validation (15%) of the networks, while the remaining 15% was reserved for testing. Dataset 09 (32,236 chunks) was reserved for further testing of the trained networks.

C. *Neural networks*

We conducted a comparison of the following neural network architectures: RNN-LSTM, RNN-biLSTM, RNN-GRU, CNN, TCN, and EDN.

RNNs are widely used for sequential data−such as sensor readings over time−thanks to their ability to maintain a hidden state that captures information from previous steps in a sequence [47]. While traditional RNNs struggle with long-term dependencies due to the vanishing gradient problem, advanced architectures like LSTM networks and GRUs address this issue through gating mechanisms [48]. GRUs offer a more compact formulation than LSTMs, while bidirectional LSTMs process sequences in both forward and reverse directions, capturing past and future context [49]. The RNNs utilized in this study are inspired by applications in voice activity detection [50] and consist of four layers (Fig. 4A): (i) a sequence input layer, (ii) an LSTM or biLSTM or

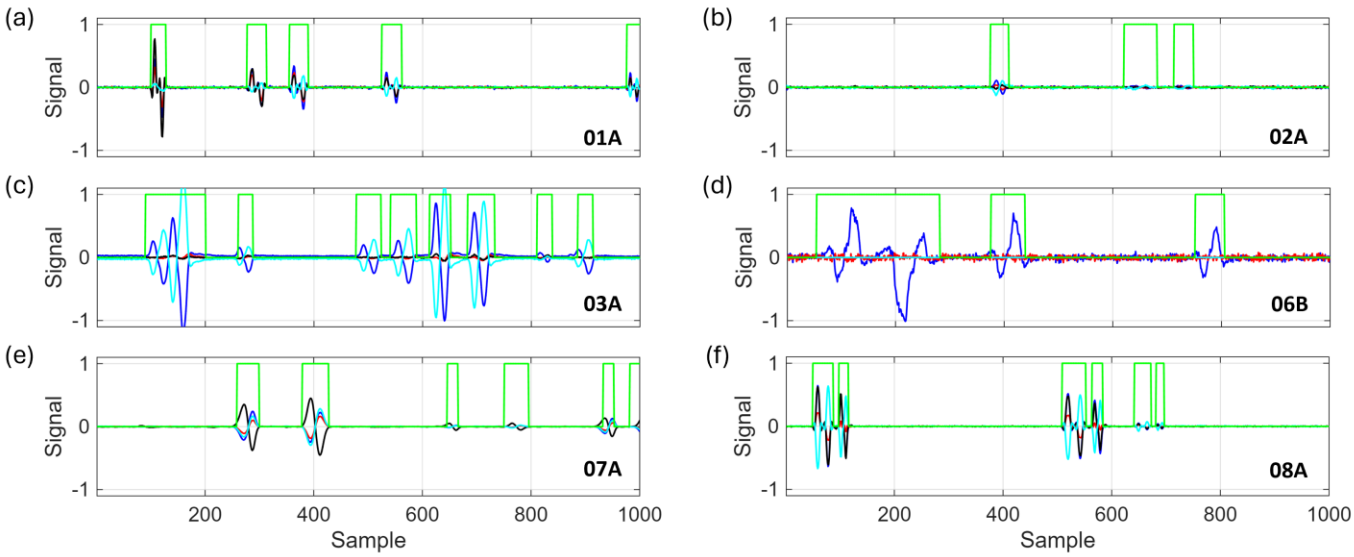

Fig. 3. Six representative examples of event-rich chunks from different datasets (blue, real signal at f_L; red, imaginary signal at f_L; black, real signal at f_H; cyan, imaginary signal at f_H; green, target segmentation mask).

GRU layer, with 64 hidden units, (iii) a fully connected layer, and (iv) a softmax layer. In previous works, we employed the biLSTM version for MIC signal segmentation within a single-dataset study [28], and a closely related model with a regression layer in place of softmax for real-time feature extraction from segmented MIC event signals [27], [28].

CNNs, originally developed for 2D grid-structured data such as images, use convolutional layers to extract hierarchical features and efficiently learn spatial patterns [47]. Their 1D counterparts are suitable for sequential data when the objective is to capture local patterns rather than long-range dependencies. A CNN with simple feed-forward structure is considered here, composed of (Fig. 4B): (i) a sequence input layer, (ii) three repeated units, each consisting of a convolutional layer, a batch normalization layer, and a rectified linear unit (ReLU) activation layer, (iii) a fully connected layer, and (iv) a softmax layer. We previously used a 2D version of this architecture to process impedance images for dielectric characterization or coincidence classification [28].

TCNs extend convolutional models by enforcing temporal causality and capturing long-range dependencies [34], [51]. They rely on causal and dilated convolutions, which expand the receptive field, and are particularly effective for sequence-to-sequence prediction. The TCN network architecture adopted in the present work is structured as follows [52] (Fig. 4C): (i) a sequence input layer, (ii) four residual blocks of dilated causal convolutions (Res DCC), with dilation factors 1, 2, 4, and 8, respectively, (iii) a fully connected layer, and (iv) a softmax layer. Each Res DCC block contains two sets of dilated causal convolution layers with the same dilation factor, followed by normalization, ReLU activation (only for the second set), and spatial dropout (with 0.005 probability) layers. A convolution layer is included in the skip connection.

Finally, an EDN architecture is considered (Fig. 4D), inspired by applications in earthquake detection [35]. The encoder begins with a down-sampling section composed of convolutional and max-pooling layers. The resulting features are then transformed into high-level representations through two residual convolutional blocks (Res CNN), a biLSTM block augmented with a Network-in-Network (NiN) module, and an additional LSTM layer. A transformer-based attention block is placed at the end of the encoder, which focuses the network on the components associated with the single-cell event signal. The decoder—consisting of an up-sampling section, a fully connected layer, and a softmax layer—maps the high-level features to a sequence of probabilities representing the presence of an event signal. Additional details of the EDN architecture are reported in Section S3 of the Supplementary Material.

In all models, the sequence input layer receives the 4-channel impedance chunks, while the scores returned by the softmax layer are used to produce a binary sequence, i.e., the predicted segmentation mask. Events are subsequently extracted from this binary mask by identifying all contiguous sequences of samples labeled as '1'. Each continuous run of ones is interpreted as an event, with its onset and offset defined by the first and last sample of the sequence, respectively. To ensure robustness, isolated single-sample

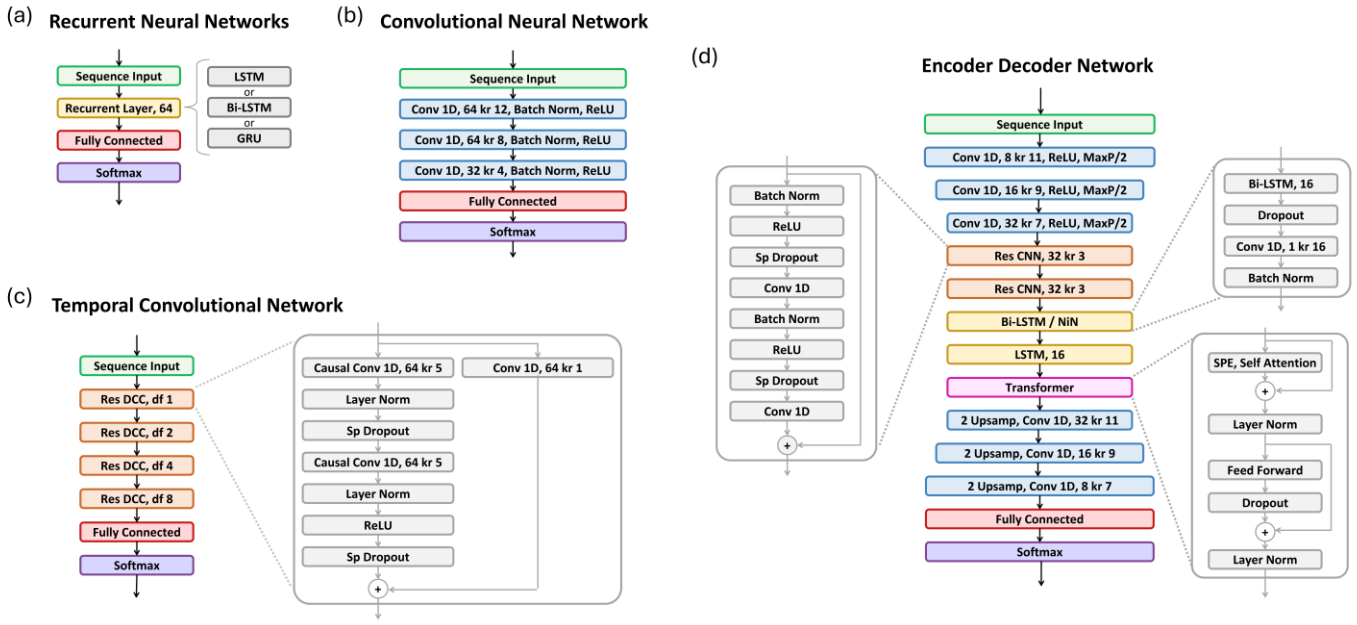

Fig. 4. Neural network architectures: (a) recurrent, (b) convolutional, (c) temporal convolutional, and (d) encoder–decoder networks. In panel (c) [resp. (d)], only one Res DCC [resp. Res CNN] block is expanded for clarity. The number of hidden units is indicated for recurrent layers, whereas the convolutional layers read as [number of kernels] kr [kernel size]. Abbreviations: Conv, convolution; Norm, normalization; df, dilation factor; Sp, spatial; MaxP/2, maxpooling with size 2 and stride 2; SPE, sinusoidal position encoding; 2 Upsamp, upsampling with factor 2. Further details of the EDN (panel (d)): all dropout probabilities are set to 0.1; the SPE layer of the transformer has dimension 16; the self-attention layer has 2 heads and 16 key channels; the feed forward unit is composed by two fully connected layers (of 64 and 16 neuros, respectively) separated by ReLU activation and followed by batch normalization.

activations and sequences shorter than a predefined minimum duration (i.e., $\delta_{ref}/4$) are discarded as spurious detections.

III. Results

A. Training, validation, and testing

The analysis was performed in the MATLAB R2025a environment. For network training, we fixed the optimization algorithm (Adam), the learning rate (0.001), the gradient threshold (1), the number of epochs (50), the batch size (128), the validation frequency (every 50 training iterations), and the loss function (cross-entropy) across all models. The networks were trained on an Intel(R) Xeon(R) w7-3445 (2.59 GHz) workstation with 192 GB of RAM, using a Nvidia RTX 2000 Ada (16 GB) GPU. The configurations achieving the best validation score were saved for testing.

Table II provides the number of trainable parameters N_P and the training time T_T of each network, along with their testing performance, i.e., the overall 0/1 classification accuracy A, the sensitivity S, and the positive predictive value PPV. S and PPV were computed as:

$$S = \frac{TP}{TP + FN}, \quad (1)$$

$$PPV = \frac{TP}{TP + FP}, \quad (2)$$

where TP, FN, and FP denote the number of correctly detected events (true positives), missed events (false negatives), and noise regions misinterpreted as events (false positives), respectively. A tolerance of $\delta_{ref}/2$ was allowed between the target and predicted event centers.

Table II. Network performance comparison. Testing results relevant to the collection of datasets 01−08.

Network	N_P	T_T (h:min)	A (%)	S (%)	PPV (%)
RNN-LSTM	17.7 k	6:33	96.9	82.8	73.2
RNN-biLSTM	35.5 k	7:46	97.9	87.0	77.9
RNN-GRU	13.3 k	8:12	96.8	82.5	74.8
CNN	44.5 k	7:06	97.0	87.8	75.7
TCN	146.6 k	7:36	97.4	86.1	76.5
EDN	39.7 k	8:58	**98.5**	**91.6**	**91.8**

(N_P, number of trainable parameters; T_T, training time).

The training time T_T (h:min) was found to range between 6:33 (RNN-LSTM) and 8:58 (EDN). Faster training could be achieved by reducing the validation frequency (e.g., to every 100 iterations) or increasing the learning rate (e.g., to 0.01), although this may lead to oscillations during training.

The minimum classification accuracy (96.8%) is obtained with the RNN-GRU, which also exhibits the lowest sensitivity (82.5%). The RNN-LSTM achieves the minimum PPV, equal to 73.2%. The best performance across all metrics is obtained with the EDN, which reaches accuracy, sensitivity, and PPV values of 98.5%, 91.6%, and 91.8%, respectively. This represents a substantial improvement over the other models. Notably, the second-best sensitivity and PPV values are 87.8% (CNN) and 77.9% (RNN-biLSTM), respectively.

Fig. 5 provides additional details on the testing performance of the EDN. Panels (a) and (b) show the distribution of sensitivity and PPV across the various datasets, respectively. The lowest values occur in dataset 03A (S = 83.7%, and PPV = 86.3%), while the highest values are observed in dataset 06A (S = 99.3%, and PPV = 98.6%). Overall, the EDN maintains good event-detection performance across all datasets 01–08, confirming its cross-dataset robustness. The capability of the network to accurately predict event length (i.e., event duration in sample) is also evaluated: panel (c) compares the histograms of predicted and target event lengths, while panel (d) reports the density plot of predicted versus target length. The correlation coefficient is $R = 0.95$ and the relative error is μ_E = 0.032 [0.015, 0.073], reported as median [interquartile range].

An analysis of how cross-setup variability affects models' performance is reported in Section S4 of the Supplementary Material.

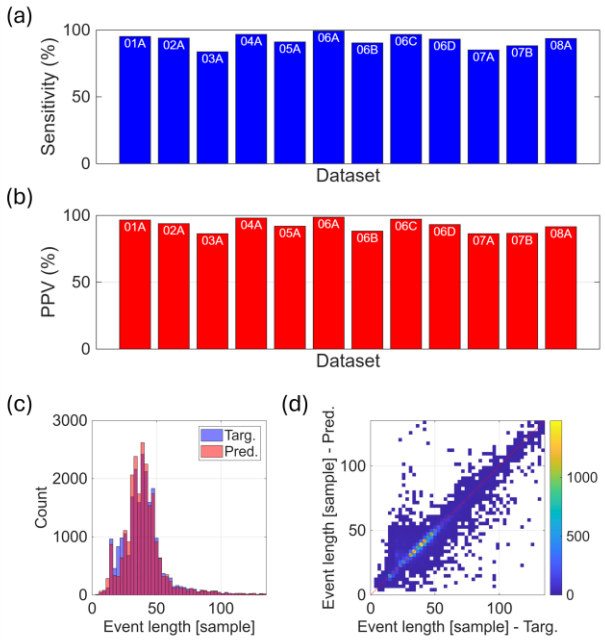

Fig. 5. Performance of the EDN network. (a) Sensitivity and (b) PPV dataset-per-dataset. (c) Histogram of target (blue) and predicted (red) event length. (d) Density plot of predicted against target event lengths. The bisector line is shown in red.

B. Additional testing

To further assess the network generalization capabilities, we performed additional testing on dataset 09 (from [41]). None of its chunks were included in the database used during the development phase. This dataset corresponds to an experimental setup employing the most common MIC chip layout, namely the standard differential measurement using three coplanar electrodes (as in Fig. 1(b)). Sub-dataset 09B is more challenging than sub-dataset 09A, as the corresponding experimental sample contained small bacteria in addition to

RBCs. Each sub-dataset is made of 16,118 chunks.

Table III reports the performance metrics (A, S, and PPV) achieved by the various networks. The highest performance metrics were achieved by the RNN-biLSTM and the EDN models. Specifically, on dataset 09A, the RNN-biLSTM achieved 99.7% accuracy, 96.3% sensitivity, and 96.0% PPV, whereas the EDN achieved 99.6% accuracy, 95.8% sensitivity, and 98.0% PPV. On dataset 09B, while the RNN-biLSTM slightly outperforms the EDN in accuracy (98.6% *vs* 98%) and PPV (77.1% *vs* 74.3%), the EDN exhibits significantly higher sensitivity (93.8% *vs* 77.9%).

Table III. Testing results for dataset 09.

Network	ID	A (%)	S (%)	PPV (%)
RNN-LSTM	09A	99.1	86.7	69.3
	09B	97.2	82.6	53.4
RNN-biLSTM	09A	**99.7**	**96.3**	96.0
	09B	**98.6**	77.9	**77.1**
RNN-GRU	09A	99.0	64.6	59.6
	09B	96.9	85.3	49.1
CNN	09A	99.2	95.3	78.5
	09B	93.5	88.7	26.2
TCN	09A	99.2	91.6	73.4
	09B	94.5	90.9	32.1
EDN	09A	99.6	95.8	**98.0**
	09B	98.0	**93.8**	74.3

IV. Discussion

In this study we investigated the ability of DL models to segment event signals from raw MIC data streams with diverse characteristics (e.g., event-signal shape, particle throughput, particle type). We trained and tested the models on data from real-world MIC experiments; signals include typical imperfections such as electronic noise, baseline drift, and occasional coincident events. This approach ensures that the trained models are immediately relevant and applicable to practical MIC scenarios.

In evaluating segmentation performance, we focus on event-detection metrics (sensitivity and PPV) rather than overall 0/1 classification accuracy. Given the class imbalance in MIC data streams (long stretches of baseline between short cell events), a high overall accuracy might not be informative. Instead, sensitivity and PPV directly assess the model's ability to correctly detect events while avoiding false alarms. In practical MIC workflows, sensitivity is especially critical (e.g., rare cell analysis), since missing an event (a false negative) means losing information about a cell, which cannot be recovered later. By contrast, a moderate number of false positives (lower PPV) is less critical, since spurious detections can be identified and filtered out in subsequent analysis stages.

Among the developed models, the EDN achieved the highest overall performance. It attained approximately 92% sensitivity and PPV on the test data, outperforming the other architectures by a noticeable margin, and it also accurately predicted the event length. The EDN maintained high performance even when evaluated on an entirely new dataset, whereas some of the simpler models suffered marked drops in performance, especially in PPV. This confirms that the EDN's structure is well-suited to capture the varied features of MIC waveforms, and that our multi-dataset training strategy successfully yielded a model that is not overfit to any single configuration.

The RNN-biLSTM, which processes the time series in both forward and reverse directions, also proved more robust than the unidirectional RNNs and achieved the second-best performance overall.

The TCN, despite having the most parameters (146.6k), did not outperform simpler models. Its training and validation metrics were closely aligned (training accuracy 98.4% vs. validation accuracy 97.5%), suggesting no evidence of overfitting. Instead, we believe the performance gap between the TCN and the EDN arises from architectural differences. The convolutional receptive field of the TCN grows in a fixed, hierarchical manner and may not optimally capture the non-uniform temporal dependencies that characterize MIC signals. In contrast, the EDN's attention mechanism can dynamically weight the most informative samples, effectively focusing the model capacity on the relevant portions of the input. This ability to allocate representational power more selectively may explain why the EDN achieves better performance despite having fewer parameters [53], [54].

An advantage of the proposed DL approach is its computational efficiency at runtime. Once trained, the networks can process new data very quickly: with a mini batch size of 128 chunks, the inference time for a single data chunk is on the order of 1–2 ms (see Section S5 of the Supplementary Material). Regarding the preprocessing steps, while baseline subtraction and normalization have negligible computational cost, the resampling step can increase the overall inference time. However, resampling can be omitted if the flow rate Φ and the sampling frequency f_s of the experimental setup are configured such that the median peak-to-peak distance δ is close to the reference value $\delta_{ref} = 16$.

With a sampling frequency $f_s = 115$ kHz, the acquisition time of 128 chunks is about 1.1 seconds. Assuming an average of 2.5 events per chunk and an inference time of 1.5 ms per chunk, the acquisition throughput and the processing throughput are 290 and 1670 events per second, respectively. Those figures are compatible with a near real-time implementation (e.g., for fast visualization of the segmented data streams). With dedicated hardware solutions, deterministic real-time implementations could be obtained, for applications such as impedance-activated sorting of single cells.

The 74,730 chunks from datasets $01-08$ already contain approximately 188,000 events, which constitute a comprehensive database representative of typical MIC scenarios. The database could be easily expanded further, e.g., extending the duration of the data streams sourced from the previous works, implementing data augmentation strategies [28], or including additional datasets from other experimental setups or synthetically generated [27]. Training databases specifically tailored for extreme MIC settings (e.g., unusually high occurrence of coincidences or signal quality significantly

below standard) could be also devised.

A limitation of the present framework comes from the supervised training approach. The latter relied on labels (i.e., target segmentation masks) generated by a traditional signal-processing algorithm, since no hand-annotated ground truth was available. These automatically generated labels are inherently imperfect – commonly referred to as noisy labels – and can contain missed events, spurious detections, and boundary inaccuracies. Moreover, when target labels are generated by a reference algorithm, there is the risk of bootstrapping bias (i.e., learning the systematic mistakes of the reference algorithm). In our study, two aspects mitigate these issues: (i) in the preparation of the database, we dedicated a large effort to fine tuning the parameters of the reference algorithm on a per-dataset base, by manually checking the algorithm's performance; (ii) our multi-dataset training across heterogeneous setups reduces the chance that the network overfits to dataset-specific artifacts of the reference algorithm. To support this, we added a manual audit of 100 chunks randomly selected from datasets 01−08, and we investigated the performance of the reference algorithm and of the EDN (see Section S6 of the Supplementary Material). The analysis showed that the label noise was limited to a small fraction of events (~10%) and that there are cases where the EDN provides better prediction than the reference algorithm. As future work, to further mitigate bootstrapping bias and label noise, we plan to implement established robust learning strategies, including noise-robust loss functions, label smoothing, annotation refinement through active data curation, and deriving the training labels by aggregating the predictions of multiple reference algorithms [55].

Beyond segmentation, the integration of MIC and DL opens further opportunities for advanced single-cell analysis. The segmentation framework developed in this work can serve as the foundation for end-to-end processing pipelines that also perform feature extraction and classification of each detected event. Specifically, the EDN architecture could be extended to such a multi-task structure, inspired by previous work [35]. This development would pave the way toward a comprehensive ML-driven MIC platform capable of detecting, analyzing, and sorting cells in real time.

V. Conclusions

This study demonstrates for the first time the capability of neural networks to interpret raw MIC data streams acquired from diverse experimental setups, addressing the challenge of cross-setup variability. In particular we addressed the segmentation task, which directly influences all subsequent processing steps. Among the developed networks, the encoder-decoder model exhibited excellent robustness with respect to signal heterogeneity arising from variations in chip layout and wiring schemes. Future work will focus on incorporating additional DL strategies for feature extraction and cell characterization. Ultimately, this approach could transform MIC into a more powerful tool for applications ranging from rapid diagnostics to high-throughput drug screening, by enabling fully automated, real-time interpretation of complex impedance data streams.

VI. Author Contribution

FC conceived the study. FC, PB, and MR designed and implemented the methodology and conducted the analyses. FC and MR prepared the original manuscript draft. All authors contributed to the development of the database and critically revised the manuscript.

References

[1] C. Honrado, P. Bisegna, N. S. Swami, and F. Caselli, "Single-cell microfluidic impedance cytometry: from raw signals to cell phenotypes using data analytics," *Lab Chip*, vol. 21, no. 1, pp. 22–54, 2021, doi: 10.1039/D0LC00840K.

[2] S. Gawad, L. Schild, and Ph. Renaud, "Micromachined impedance spectroscopy flow cytometer for cell analysis and particle sizing," *Lab Chip*, vol. 1, no. 1, pp. 76–82, 2001, doi: 10.1039/B103933B.

[3] T. Sun and H. Morgan, "Single-cell microfluidic impedance cytometry: a review," *Microfluid. Nanofluid.*, vol. 8, no. 4, pp. 423–443, 2010, doi: 10.1007/s10404-010-0580-9.

[4] M. Shaker, L. Colella, F. Caselli, P. Bisegna, and P. Renaud, "An impedance-based flow micro-cytometer for single cell morphology discrimination," *Lab Chip*, vol. 14, no. 14, pp. 2548–2555, 2014, doi: 10.1039/c4lc00221k.

[5] T. Tang *et al.*, "Microscopic impedance cytometry for quantifying single cell shape," *Biosens. Bioelectron.*, vol. 193, 2021, doi: 10.1016/j.bios.2021.113521.

[6] R. Reale, A. De Ninno, T. Nepi, P. Bisegna, and F. Caselli, "Extensional-Flow Impedance Cytometer for Contactless and Optics-Free Erythrocyte Deformability Analysis," *IEEE Trans. Biomed. Eng.*, vol. 70, no. 2, pp. 565–572, Oct. 2023, doi: 10.1109/TBME.2022.3197214.

[7] J. Jarmoshti *et al.*, "Neural Network-Enabled Multiparametric Impedance Signal Templating for High throughput Single-Cell Deformability Cytometry Under Viscoelastic Extensional Flows," *Small*, vol. n/a, no. n/a, p. 2407212, doi: https://doi.org/10.1002/smll.202407212.

[8] J. Chen, X. Zou, D. C. Spencer, and H. Morgan, "Single-cell electro-mechanical shear flow deformability cytometry," *Microsyst. Nanoeng.*, vol. 10, no. 1, p. 173, 2024, doi: 10.1038/s41378-024-00810-5.

[9] Y. Feng, H. Chai, W. He, F. Liang, Z. Cheng, and W. Wang, "Impedance-Enabled Camera-Free Intrinsic Mechanical Cytometry," *Small Methods*, vol. 6, no. 7, Oct. 2022, doi: 10.1002/smtd.202200325.

[10] R. Reale, A. De Ninno, L. Businaro, P. Bisegna, and F. Caselli, "Electrical measurement of cross - sectional position of particles flowing through a microchannel," *Microfluid. Nanofluidics*, vol. 22, no. 41, pp. 1–13, 2018, doi: 10.1007/s10404-018-2055-3.

[11] H. Daguerre, M. Solsona, J. Cottet, M. Gauthier, P. Renaud, and A. Bolopion, "Positional dependence of particles and cells in microfluidic electrical impedance flow cytometry: origin, challenges and opportunities," *Lab Chip*, vol. 20, no. 20, pp. 3665–3689, 2020, doi: 10.1039/d0lc00616e.

[12] M. Righetto, C. Brandi, R. Reale, and F. Caselli, "Integrating impedance cytometry with other microfluidic tools towards multifunctional single-cell analysis platforms," *Lab Chip*, vol. 25, no. 5, pp. 1316–1341, 2025, doi: 10.1039/D4LC00957F.

[13] F. Dadkhah Tehrani, M. D. O'Toole, and D. J. Collins, "Tutorial on impedance and dielectric spectroscopy for single-cell characterisation on microfluidic platforms: theory, practice, and recent advances," *Lab Chip*, vol. 25, no. 5, pp. 837–855, 2025, doi: 10.1039/D4LC00882K.

[14] B. de Wagenaar *et al.*, "Towards microfluidic sperm refinement: impedance-based analysis and sorting of sperm cells," *Lab Chip*, vol. 16, no. 8, pp. 1514–1522, 2016, doi: 10.1039/C6LC00256K.

[15] M. Evander, A. J. Ricco, J. Morser, G. T. A. Kovacs, L. L. K. Leung, and L. Giovangrandi, "Microfluidic impedance cytometer for platelet analysis," *Lab Chip*, vol. 13, no. 4, pp. 722–729, 2013, doi: 10.1039/c2lc40896a.

[16] T. Sun, C. van Berkel, N. G. Green, and H. Morgan, "Digital signal processing methods for impedance microfluidic cytometry," *Microfluid. Nanofluid.*, vol. 6, no. 2, pp. 179–187, 2009, doi: 10.1007/s10404-008-0315-3.

[17] F. Caselli and P. Bisegna, "A simple and robust event-detection algorithm for single-cell impedance cytometry," *IEEE Trans Biomed Eng*, vol. 63, no. 2, pp. 415–422, 2016, doi: 10.1109/TBME.2015.2462292.

[18] R. Liu, N. Wang, F. Kamili, and A. F. Sarioglu, "Microfluidic CODES: a scalable multiplexed electronic sensor for orthogonal detection of particles in microfluidic channels," *Lab Chip*, vol. 16, no. 8, pp. 1350–1357, 2016, doi: 10.1039/C6LC00209A.

[19] B. Wurts *et al.*, "A Derivative-Based Framework for Real-Time Signal Processing and Event Detection in Impedance Flow Cytometry," *Sensors*, vol. 25, no. 23, 2025, doi: 10.3390/s25237252.

[20] T. Tang *et al.*, "A review on intelligent impedance cytometry systems: Development, applications and advances," 2023. doi: 10.1016/j.aca.2023.341424.

[21] Y. Feng *et al.*, "Neural Network Enhanced Real-Time Impedance Flow Cytometry for Single-Cell Intrinsic Characterization," *Lab Chip*, no. October, p., 2021, doi: 10.1039/D1LC00755F.

[22] M. Daorazio *et al.*, "Electro-Optical Classification of Pollen Grains via Microfluidics and Machine Learning," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 921–931, Oct. 2022, doi: 10.1109/TBME.2021.3109384.

[23] C. Honrado, A. Salahi, S. J. Adair, J. H. Moore, T. W. Bauer, and N. S. Swami, "Automated biophysical classification of apoptotic pancreatic cancer cell subpopulations by using machine learning approaches with impedance cytometry," *Lab Chip*, p., 2022, doi: 10.1039/D2LC00304J.

[24] B. K. Ashley, J. Sui, M. Javanmard, and U. Hassan, "Multi-modal sensing with integrated machine learning to differentiate specific leukocytes targeted by electrically sensitive hybrid particles," *Biosens. Bioelectron.*, vol. 241, p. 115661, 2023, doi: https://doi.org/10.1016/j.bios.2023.115661.

[25] S. Zhang *et al.*, "Convolutional Neural Network-Driven Impedance Flow Cytometry for Accurate Bacterial Differentiation," *Anal. Chem.*, vol. 96, no. 11, pp. 4419–4429, Mar. 2024, doi: 10.1021/acs.analchem.3c04421.

[26] J. Wei *et al.*, "Machine learning classification of cellular states based on the impedance features derived from microfluidic single-cell impedance flow cytometry," *Biomicrofluidics*, vol. 18, no. 1, 2024, doi: 10.1063/5.0181287.

[27] C. Honrado, J. S. McGrath, R. Reale, P. Bisegna, N. S. Swami, and F. Caselli, "A neural network approach for real-time particle/cell characterization in microfluidic impedance cytometry," *Anal. Bioanal. Chem.*, vol. 412, no. 16, pp. 3835–3845, 2020, doi: 10.1007/s00216-020-02497-9.

[28] F. Caselli, R. Reale, A. De Ninno, D. Spencer, H. Morgan, and P. Bisegna, "Deciphering impedance cytometry signals with neural networks," *Lab Chip*, vol. 22, no. 9, pp. 1714–1722, 2022, doi: 10.1039/D2LC00028H.

[29] N. Wang, R. Liu, N. Asmare, C. H. Chu, and A. F. Sarioglu, "Processing code-multiplexed Coulter signals: Via deep convolutional neural networks," *Lab Chip*, vol. 19, no. 19, pp. 3292–3304, 2019, doi: 10.1039/c9lc00597h.

[30] N. Wang, R. Liu, N. Asmare, C.-H. Chu, O. Civelekoglu, and A. F. Sarioglu, "Closed-loop feedback control of microfluidic cell manipulation via deep-learning integrated sensor networks," *Lab Chip*, vol. 21, pp. 1916–1928, 2021, doi: 10.1039/D1LC00076D.

[31] M. Baur, M. Reisbeck, O. Hayden, and W. Utschick, "Joint Particle Detection and Analysis by a CNN and Adaptive Norm Minimization Approach," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 8, 2022, doi: 10.1109/TBME.2022.3147701.

[32] K. Cheung, S. Gawad, and P. Renaud, "Impedance spectroscopy flow cytometry: on-chip label-free cell differentiation," *Cytometry Part A*, vol. 65A, no. 2, pp. 124–132, 2005, doi: 10.1002/cyto.a.20141.

[33] N. Demierre, T. Braschler, P. Linderholm, U. Seger, H. Van Lintel, and P. Renaud, "Characterization and optimization of liquid electrodes for lateral dielectrophoresis," *Lab Chip*, vol. 7, no. 3, pp. 355–365, 2007, doi: 10.1039/B612866A.

[34] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *CoRR*, vol. abs/1803.01271, 2018, [Online]. Available: http://arxiv.org/abs/1803.01271

[35] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nat. Commun.*, vol. 11, no. 1, p. 3952, 2020, doi: 10.1038/s41467-020-17591-w.

[36] D. Spencer and H. Morgan, "High-Speed Single-Cell Dielectric Spectroscopy," *ACS Sens.*, vol. 5, no. 2, pp. 423–430, 2020, doi: 10.1021/acssensors.9b02119.

[37] A. De Ninno *et al.*, "High-throughput label-free characterization of viable, necrotic and apoptotic human lymphoma cells in a coplanar-electrode microfluidic impedance chip," *Biosens. Bioelectron.*, vol. 150, p. 111887, 2020, doi: 10.1016/j.bios.2019.111887.

[38] F. Caselli, A. De Ninno, R. Reale, L. Businaro, and P. Bisegna, "A novel wiring scheme for standard chips enabling high-accuracy impedance cytometry," *Sens. Actuators B Chem.*, vol. 256, pp. 580–589, 2018, doi: 10.1016/j.snb.2017.10.113.

[39] C. Brandi *et al.*, "On the compatibility of single-cell microcarriers (nanovials) with microfluidic impedance cytometry," *Lab Chip*, vol. 24, no. 11, pp. 2883–2892, Oct. 2024, doi: 10.1039/d4lc00002a.

[40] R. Reale, A. De Ninno, L. Businaro, P. Bisegna, and F. Caselli, "A simple electrical approach to monitor dielectrophoretic focusing of particles flowing in a microchannel," *Electrophoresis*, vol. 40, no. 10, pp. 1400–1407, 2019, doi: 10.1002/elps.201800423.

[41] C. Troiano *et al.*, "Rapid Assessment of Susceptibility of Bacteria and Erythrocytes to Antimicrobial Peptides by Single-Cell Impedance Cytometry," *ACS Sens.*, vol. 8, no. 7, pp. 2572–2582, Jul. 2023, doi: 10.1021/acssensors.3c00256.

[42] F. Caselli, A. De Ninno, R. Reale, L. Businaro, and P. Bisegna, "A Bayesian Approach for Coincidence Resolution in Microfluidic Impedance Cytometry," *IEEE Trans Biomed Eng*, vol. 68, no. 1, pp. 340–349, 2020, doi: 10.1109/TBME.2020.2995364.

[43] M. R. Kellman, F. R. Rivest, A. Pechacek, L. L. Sohn, and M. Lustig, "Node-Pore Coded Coincidence Correction: Coulter Counters, Code Design, and Sparse Deconvolution," *IEEE Sens. J.*, vol. 18, no. 8, pp. 3068–3079, 2018, doi: 10.1109/JSEN.2018.2805865.

[44] Y. Xia, J. Guo, Y. Shi, G. Jiang, Z. Gu, and H. Wang, "A Sparse Constrained Optimization Method for Resolving Coincident Single-Cell Events in Microfluidic-Based Impedance Sensing," *IEEE Trans. Biomed. Eng.*, 2025, doi: 10.1109/TBME.2025.3643493.

[45] S. Li, C. Wang, B. Yang, X. Liang, and J. Li, "An Effective Recognition Method for Particle Coincidence in Double Differential Impedance Cytometry," *IEEE Sens. J.*, vol. 23, no. 16, 2023, doi: 10.1109/JSEN.2023.3294012.

[46] M. Leuthner, R. Vorländer, and O. Hayden, "Disentangling Coincident Cell Events Using Deep Transfer Learning and Compressive Sensing," *Advanced Intelligent Systems*, 2025, doi: 10.1002/aisy.202500766.

[47] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning : The MIT Press, 2016, 800 pp, ISBN: 0262035618," *Genet. Program. Evolvable Mach.*, vol. 19, no. 1–2, 2017.

[48] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014. [Online]. Available: https://arxiv.org/abs/1412.3555

[49] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks," 2016. [Online]. Available: https://arxiv.org/abs/1609.08409

[50] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with LSTM Recurrent Neural Networks and an application to Hollywood movies," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 483–487. doi: 10.1109/ICASSP.2013.6637694.

[51] T. Dissanayake, T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Multi-stage stacked temporal convolution neural networks (MS-S-TCNs) for biosignal segmentation and anomaly localization," *Pattern Recognit.*, vol. 139, 2023, doi: 10.1016/j.patcog.2023.109440.

[52] Mathworks, "Sequence-to-Sequence Classification Using 1-D Convolutions," https://it.mathworks.com/help/deeplearning/ug/sequence-to-sequence-classification-using-1-d-convolutions.html.

[53] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017. doi: 10.1201/9781003561460-19.

[54] X. Wei and Z. Wang, "TCN-attention-HAR: human activity recognition based on attention mechanism time convolutional network," *Sci. Rep.*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-57912-3.

[55] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis," *Med. Image Anal.*, vol. 65, 2020, doi: 10.1016/j.media.2020.101759.

[56] X. Zou, "Single-cell Impedance Analysis and Sorting," University of Southampton, Faculty of Engineering and Physical Science, Electronics and Computer Science, 2025.