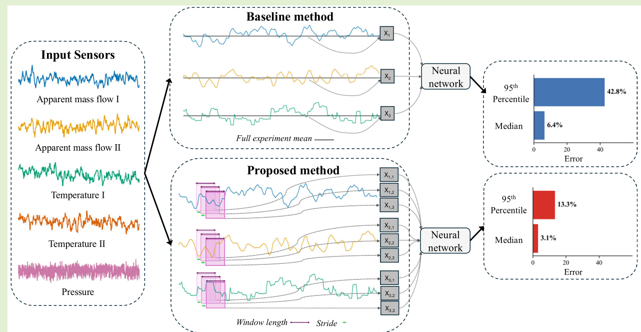
Temporal Data and Short-Time Averages Improve Multiphase Mass Flow Metering

Amanda Nyholm, Yessica Arellano, Jinyu Liu, Damian Krakowiak, and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—Reliable flow measurements are essential in many industries, but current instruments often fail to accurately estimate multiphase flows, which are frequently encountered in real-world operations. Combining machine learning (ML) algorithms with accurate single-phase flowmeters has therefore received extensive research attention in recent years. The Coriolis mass flowmeter (CMF) is a widely used single-phase meter that provides direct mass flow measurements, which ML models can be trained to correct, thereby reducing measurement errors in multiphase conditions. This article demonstrates that preserving temporal information significantly improves model performance in such scenarios. We compare a multilayer perceptron (MLP), a windowed MLP, a long short-term memory (LSTM) model, and a convolutional neural network (CNN) on three-phase air–water–oil flow data from 342 experiments. Whereas prior work typically compresses each experiment into a single averaged sample, we instead compute short-time averages from within each experiment and train models that preserve temporal information at several downsampling intervals. The CNN performed best at 0.25 Hz with approximately 95% of relative errors (REs) below 13%, a normalized root-mean-squared error (MSE) of 0.03, and a mean absolute percentage error of approximately 4.3%, clearly outperforming the best single-averaged model and demonstrating that short-time averaging within individual experiments is preferable. While the CNN performed best, all three sequence models outperformed the nontemporal baseline MLP, clearly illustrating the benefits of temporal data and short-time averages. Results are consistent across multiple data splits and random seeds, demonstrating robustness.**



***Index Terms*—Convolutional neural networks (CNNs), Coriolis mass flowmeters (CMFs), mass flow estimation, multiphase flow measurement, multivariate time series.**

I. Introduction

ACCURATE measurements of mass flow in mixtures are essential in several industries ranging from *carbon capture and storage* (CCS) [1] to the food and beverage [2] and oil and gas [3] sectors. These measurements underpin fiscal metering, process control, regulatory compliance, and environmental monitoring, among other applications. Mass flow can be estimated using various meter types, and the choice is often application-dependent.

Received 26 February 2026; accepted 7 April 2026. Date of publication 17 April 2026; date of current version 2 June 2026. This work was supported by the Project PREFERENCE under Project 336355. The associate editor coordinating the review of this article and approving it for publication was Prof. Chao Tan. *(Corresponding author: Amanda Nyholm.)*

Amanda Nyholm is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: amanda.nyholm@ntnu.no).

Yessica Arellano is with the Department of Gas Technology, SINTEF Energy Research, 7465 Trondheim, Norway (e-mail: yessica.arellano@sintef.no).

Jinyu Liu and Damian Krakowiak are with the Department of Research and Development, KROHNE Ltd., NN8 6AE Wellingborough, U.K. (e-mail: j.liu4@krohne.com; d.krakowiak@krohne.com).

Pierluigi Salvo Rossi is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway, and also with the Department of Gas Technology, SINTEF Energy Research, 7465 Trondheim, Norway (e-mail: salvorossi@ieee.org).

Digital Object Identifier 10.1109/JSEN.2026.3683143

Venturi meters (VMs) are differential-pressure meters widely used in chemical as well as oil and gas industries to estimate wet gases. They do not measure mass flow rate directly; instead, they measure pressure differences, from which the volumetric flow rate is inferred using Bernoulli's principle and then converted to mass flow rate using the fluid density [4].

Coriolis mass flowmeters (CMFs) are the only commercial meters that provide direct, highly accurate measurements of both mass flow and fluid density [5]. Mass flow is estimated by exploiting inertial forces generated as the fluid passes through vibrating tubes, causing a phase shift proportional to the mass flow [6], while the density is determined from the tubes' natural oscillation frequency, which varies with the fluid mass [7]. CMFs also provide temperature measurements and additional internal signals, such as tube vibration amplitudes and the oscillation frequency (accessible only to the manufacturer).

The high accuracy of CMFs is restricted to single-phase flows, as they assume that the fluid moves homogeneously through the tubes [8]. In multiphase conditions, entrained gas or phase separation disturbs the tube oscillations and breaks the proportional relationship with mass flow, leading to large measurement errors, even for low amounts of gases [9]. Multiphase mass flowmeters are commercially available, but they are expensive, tend to require extensive maintenance, and some designs rely on radioactive sources [10].

In recent years, the interest in *machine learning* (ML) algorithms for multiphase mass flow estimation has increased significantly [11], [12]. Two distinct approaches have emerged: 1) using sensor outputs from a single-phase meter, possibly combined with additional sensors to apply corrections and 2) using measurements from other sensors, such as temperature and pressure, to estimate mass flow directly. The latter is often referred to as *virtual flow metering* (VFM). While these approaches are related, they address different underlying problems and are suited for different applications: correction algorithms are, for example, more relevant in industries where CMFs are widely used, whereas VFM is often applied in fields or installations where direct mass flow measurements are impractical. This article focuses on correction algorithms for CMFs under multiphase conditions.

A. Previous Work

1) Conventional Coriolis Mass Flow Metering Models: The first CMF correction algorithm was introduced in 2001 when Liu et al. [7] trained a *multilayer perceptron* (MLP) on an air–water mixture, reducing the *relative error* (RE) from ±20% to ±2% within a limited range of operating conditions. Wang et al. [13] extended this approach in 2016 with an MLP trained on liquid flow rates between 700 and 14 500 $\mathrm{kg\,h^{-1}}$ and *gas volume fractions* (GVFs)[1] up to 30%, reducing the RE from ±40% to ±1.5% in vertical pipes and ±2.5% in horizontal pipes. Both implementations used a combination of user-accessible and internal manufacturer signals as inputs.

Several studies have addressed the correction of CMF measurements in two-phase CO_2 mixtures to achieve accurate CO_2 monitoring in CCS systems. Wang et al. [14] trained an MLP with only two (user-accessible) features on liquid CO_2 flow rates ranging from 300 to 3050 $\mathrm{kg\,h^{-1}}$ and GVFs up to 90% and obtained most REs below ±1.5% (horizontal) and ±2% (vertical). Wang et al. [15] extended this work with a two-stage model on liquid CO_2 flow rates from 250 to 3200 $\mathrm{kg\,h^{-1}}$ and GVFs up to 75%, where the first stage classified the flow pattern and the second stage estimated the mass flow, yielding REs below ±5% (horizontal) and ±3% (vertical). A dynamic ensemble method, which first classified the flow pattern before estimating the mass flow using a least-squares support vector regressor and user-accessible features, achieved REs below ±1% over GVFs ranging from 3.1% to 88.4% and liquid CO_2 flow rates between 200 and 3100 $\mathrm{kg\,h^{-1}}$ [16]. Sun et al. [17] used several tree-based algorithms to identify the most important features before training a gradient boosting random forest to estimate the mass flow, achieving 70% of REs within ±1.2% across flow rates ranging from 212 to 3449 $\mathrm{kg\,h^{-1}}$ and GVFs between 1.82% and 77.29%. A recent study by Wang et al. [18] employed an MLP to correct the mass flow for liquid CO_2 injected with gaseous nitrogen and reported that most REs were within ±4%. Chowdhury et al. [19] applied CMF correction algorithms to slurry (sand–water) mixtures using Gaussian process regression, yielding notably low REs below ±0.2%, outperforming both a *support vector machine* (SVM) and an MLP.

Henry et al. [20] fed outputs from a CMF and a water cut (water content in oil) meter into an MLP to estimate the mass flow of a three-phase water–oil–nitrogen mixture. The experiments involved high-viscosity oil (200 cSt), GVFs between 4.2% and 48%, water cuts ranging from 0.2% to 98% and total liquid mass flow rates of 2.5–11.6 $\mathrm{kg\,h^{-1}}$. The final model achieved REs below ±2.5%.

2) Sequence Models in Flow Measurements: Even under stable conditions, flow measurements exhibit short-term fluctuations, providing a strong motivation for incorporating temporal dependencies into model development. Such models, known as sequence models, are uncommon in CMF corrections but have shown clear benefits in related domains [21], with *long short-term memory* (LSTM) networks proving especially effective in VFM [22].

Wang et al. [23] acknowledged the time-series nature of flow measurements in 2020 and used moving-averaged VM outputs to estimate volumetric flow rates of a three-phase oil–water–gas mixture using SVM, MLP, and *convolutional neural network* (CNN) models. Building on this, Wang et al. [24] later developed CNN–LSTM and *temporal convolutional network* (TCN) models to estimate the volumetric liquid and gas flow rates of the oil–water–gas mixture. The same research group further extended their work on the TCN by combining it with a dual-plane electrical capacitance tomography sensor [25]. Jiang et al. [26] instead combined an electrical resistance tomography sensor with a VM to estimate the volumetric flow rate of an air–water mixture using a transformer and 5-min-averaged time-series inputs, which outperformed several other sequence models.

Novel sequence deep-learning models for two-phase gas–liquid mixtures have been proposed in recent years. Bao et al. [27] fused multiple sensor signals and proposed two sequence models—one based on a *one-dimensional CNN* (1D-CNN) and an LSTM, and another incorporating positional encodings, attention, and a sliding window—for estimating both mass flow and GVF. Gao et al. [28] proposed a multitask temporal channel-wise CNN for gas void fraction estimation, using four conductance sensors and flow pattern classification as an auxiliary task. Zhang et al. [29] further advanced this line of work by proposing a dual CNN–Transformer mixture *neural network* (NN), augmented with an LSTM, for GVF estimation, which outperformed conventional CNN and transformer-based models.

Using sequence models to correct CMF measurements is a logical next step, yet few studies have investigated them. Zhang et al. [30] demonstrated promising results on

[1]GVF represents the volumetric fraction of gas in a gas–liquid mixture, where 0% is pure liquid and 100% is pure gas.

a single-phase water mixture using raw tube vibration signals from a CMF as inputs to an LSTM. Sun et al. [31] recently compared several sequence models on gas–liquid two-phase data with liquid mass flow rates between 0 and 1000 $\mathrm{kg\,h^{-1}}$ and GVFs of 0%–30%, though without a nonsequence baseline.

B. Contributions

Obtaining reliable mass flow rates for individual phases under multiphase conditions is challenging. Single-phase flowmeters are often preferred over more complex multiphase technologies, but their accuracy deteriorates significantly in multiphase flows. CMFs are attractive in multiphase flow conditions because, although their errors are large, they show repeatability. Averaging is often applied to reduce errors, but accuracy remains below industrial requirements. For example, in the dataset used in this work (described in Section II), the maximum RE of the uncorrected CMF measurements was 409%, and averaging reduced it to just 69%. These limitations motivate further research into ML-based correction algorithms.

While sequence models have been applied in prior work, the sources of their performance gains and their robustness have not been systematically established. This article addresses these gaps through controlled model comparisons, short-time averaging strategies, and extensive statistical validation on a complex three-phase flow dataset.

This work makes the following contributions.

1) We demonstrate, through controlled comparisons between three sequence models and a baseline MLP, that preserving temporal information improves CMF correction independent of model capacity.
2) We show that short-time averaging within experiments outperforms the current norm of averaging over entire experiments.
3) We establish the robustness of the results through multiple evaluation metrics, disjoint data splits, and repeated experiments with multiple random seeds.
4) We show that the results hold across a wide range of three-phase flow conditions, including GVFs up to 95%, using only user-accessible features to ensure reproducibility.

This article is organized into six sections. Section II introduces the dataset and its origin, followed by Section III, which presents the necessary mathematical preliminaries. Implementation details are given in Section IV, and the results are discussed in Section V. Finally, Section VI summarizes the work and outlines future research directions.

Notation—Vectors and matrices are denoted with bold lowercase letters and bold uppercase letters, respectively; the transpose operator is represented with $(\cdot)^\top$.

II. Experimental Setup and Data

The dataset consists of 294 153 measurements of a three-phase air–water–oil mixture, collected at 342 distinct operating points, each hereafter referred to as an experiment. An operating point is defined by a unique combination of water cut, viscosity, oil mass flow rate, total mass flow rate, and GVF. For each baseline setting (water cut, viscosity, and oil and total mass flow rates), the GVF was incremented across several consecutive experiments. Each experiment lasted about 60 s at a fixed GVF and was sampled at 14.3 Hz. Table I summarizes the ranges of the operating variables, and Fig. 1 shows selected measured and reference variables for four consecutive experiments.

TABLE I
Ranges of Operating Variables

Variable	Unit	Lower limit	Upper limit
Pressure	bar	1.01	4.49
Viscosity	mPa s	7.17×10^{-4}	666
Water cut	%	0	99.4
Temperature	°C	19.2	35.7
Oil mass flow rate	$\mathrm{kg\,h^{-1}}$	10.6	12,900
Gas volume fraction	%	0	95.5
Overall mass flow rate	$\mathrm{kg\,h^{-1}}$	930	14,900

An overview of the experimental setup is shown in Fig. 2. The multiphase skid was constructed from stainless-steel piping with a pipe diameter of 1 in. The skid was designed to separate the gas from the liquid mixture using gravity before recombining the streams, as illustrated in Fig. 3, to obtain representative measurements of the water cut. This type of short-term gas–liquid separation with subsequent remixing is common in multiphase flow measurement systems, particularly within the oil and gas industry [32]. Two CMFs were used to measure mass flow: a KROHNE OPTIMASS 6400 S25 (1 in.) CMF for the air–water–oil mixture and a KROHNE OPTIMASS 6400 S08 (1/4 in.) CMF for the oil–water mixture, each with a liquid mass flow accuracy of ±0.1% under single-phase conditions [33]. A KROHNE OPTIBAR PM 3050 pressure transmitter was installed downstream of the OPTIMASS 6400 S25. The transmitter was configured for 0–5 bar, with a reference accuracy of ±0.1% [34]. The true mass flow was determined by combining the liquid reference and air reference measurements, whereas the apparent overall mass flow rate was obtained by combining the mass flow outputs of the two CMFs in the skid.

The oil phase consisted of mixtures of Q8 Porta 12P and Q8 Porta 475P oil, prepared at two blend ratios to obtain a low-viscosity oil and a high-viscosity oil. For each blend, samples were taken from the tank and analyzed in the internal lab to determine density and viscosity over the temperature range 10°C–50°C in 5°C increments. The resulting viscosity–temperature and density–temperature curves were used to determine the reference viscosity and density at each meter location based on the measured CMF temperature. This procedure is analogous to field practice and reflects realistic operating conditions.

The water viscosity was computed from IAPWS-97 as a function of temperature (from the CMF output) and pressure (from the downstream OPTIBAR PM 3050 measurement). The oil viscosity was obtained from the measured viscosity–temperature curves. The effective mixture viscosity was estimated using the power law mixing model, which is common for layered or poorly mixed fluids.

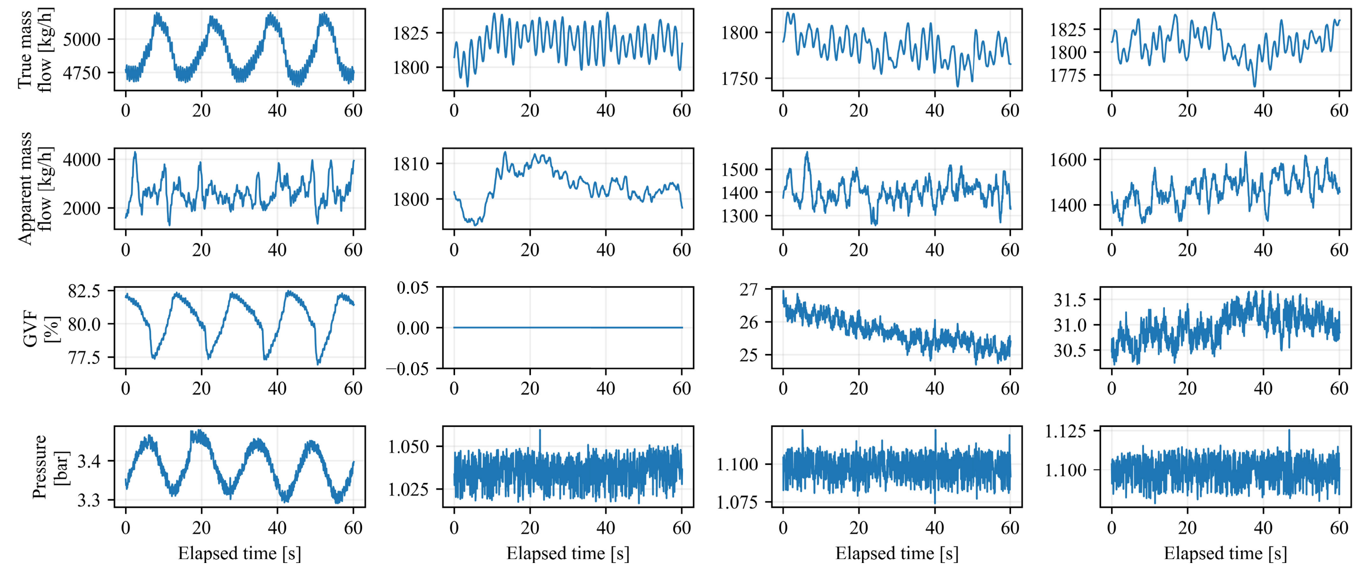


Fig. 1. Representative time series showing (from top to bottom) the true mass flow rate, apparent mass flow rate, GVF, and pressure for four consecutive experiments (columns).

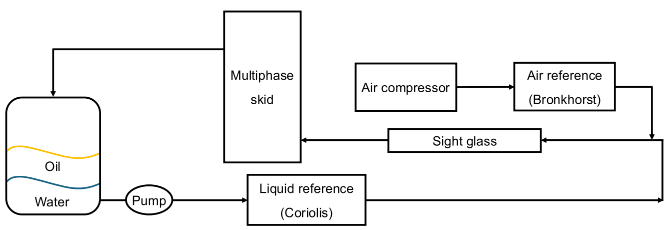


Fig. 2. Illustration of the experimental setup.

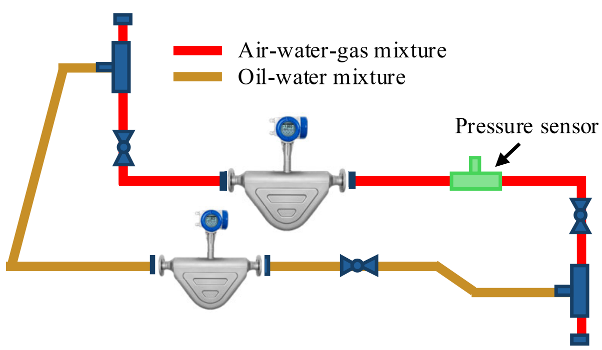


Fig. 3. Illustration of the multiphase skid, rotated 90° clockwise for compactness.

III. Preliminaries

A. Multivariate Time Series

A time series is an ordered sequence of measurements, where current values often depend on prior ones (e.g., temperature readings). Time-series modeling can be performed using either statistical methods, such as Kalman filters and ARIMA models, or ML models. It is crucial to prevent data leakage (using future values) when training a model that incorporates temporal information, as this can inflate evaluation metrics on validation and test sets. Creating training/test splits, as well as designing *cross-validation* (CV) schemes, requires more care than for models that do not incorporate temporal information.

Furthermore, ML models require specifying the number of past samples to consider, commonly referred to as the window. The window length depends on the problem and the characteristics of the data itself.

Mathematically, a multivariate time series with D variables and N consecutive measurements is defined as

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N} \tag{1}$$

$$\boldsymbol{x}_n = \left(x_{1,n}, x_{2,n}, \ldots, x_{D,n}\right)^{\top} \in \mathbb{R}^{D}, \quad n = 1, \ldots, N \tag{2}$$

where the dth row of $\boldsymbol{X}$ corresponds to the values of the dth variable and the nth column corresponds to the measurements at the nth time step.

B. Multilayer Perceptron Models

MLPs are among the simplest ML models and have been referred to under various names—including NNs, artificial NNs, deep NNs, and backpropagation-ANNs—in previous studies [7], [13], [14], [19], [23]. Despite their simplicity, MLPs have good predictive capabilities, as they combine linear operations with nonlinear activation functions ($\varphi(\cdot)$), allowing them to approximate complex relationships. Their capacity depends on the number of hidden layers (L) and nodes per layer (m_ℓ). For a scalar-output MLP, it can be written as

$$\hat{y} = \sum_{j=1}^{m_L} w_j^{(L+1)} \varphi\left(\sum_{i=1}^{m_{L-1}} w_{ji}^{(L)} \varphi(\cdots) + b_j^{(L)}\right) + b^{(L+1)} \tag{3}$$

where $w_{ji}^{(\ell)}$ and $b_j^{(\ell)}$ denote the weights and biases for the ℓth layer, respectively [35, pp. 33–35 and 41–43]. The weights are updated using gradient-based optimizers, such as stochastic gradient descent and Adam.

1) Windowed MLP: A standard MLP does not inherently account for past samples, limiting its applicability to time-series data. However, using a sliding window, past samples can be provided as inputs, enabling temporal dependencies to be captured by transforming sequences into fixed-size input vectors. We refer to this architecture as MLPw. Formally, we can express the inputs as

$$\boldsymbol{S}_n = \big[\boldsymbol{x}_{n-W+1},\ \dots,\ \boldsymbol{x}_n \tag{4}$$
$$\big] \in \mathbb{R}^{D\times W}, \quad n = W, \dots, N \tag{5}$$

where W denotes the window length. Note that $\boldsymbol{S}_n$ is reshaped into a 1-D vector before being passed to the MLPw.

C. Convolutional Neural Networks

In contrast to MLPs, CNNs capture relationships between neighboring inputs, enabling better identification of local structures, which explains their extensive use in image processing [35, p. 61]. They typically use a collection of convolutional layers to extract local patterns, followed by pooling layers to reduce dimensionality and computation, and finally fully connected layers to produce the outputs [36].

In addition to their traditional use, CNNs can also be applied to time-series data using a 1D-CNN, where the convolution is performed along the temporal domain [21]. Compared to other sequence ML models such as LSTMs and Transformers, 1D-CNNs are relatively simple.

The output of one *1-D convolutional layer* (Conv1D) at time step n is

$$h_n = \varphi\left(b + \sum_{k=1}^{K} \omega_k\, x_n + k - 1\right) \tag{6}$$

where x_n denotes the input, ω_k are the kernel weights of length K, b is the bias, and $\varphi(\cdot)$ is the nonlinear activation function [35]. Note that (6) is shown for a single input channel for simplicity; for multivariate inputs, the convolution also sums across input channels.

D. Long Short-Term Memory models

LSTMs belong to a class of *recurrent NNs* (RNNs) and are designed to model sequential data while alleviating the vanishing gradient problem in traditional RNNs [37]. Unlike feedforward models such as MLPs and CNNs, LSTMs maintain an internal memory of previous states, enabling them to capture long-term dependencies in the data.

Gating mechanisms are used to control how previous information is stored, forgotten, and used in LSTMs. An LSTM cell state at time n, c_n, is computed as

$$c_n = f_n \odot c_{n-1} + i_n \odot \tilde{c}_n, \qquad h_n = o_n \odot \tanh(c_n) \tag{7}$$

where $\tilde{c}_n$ is the candidate cell state, h_n is the hidden state, f_n, i_n, and o_n are the forget, input, and output gates, respectively, and $\odot$ is the element-wise multiplication [38]. The gated memory structure makes LSTMs well-suited for sequence tasks in which information from earlier time steps may remain relevant.

E. Evaluation Metrics

REs are commonly used to evaluate model performance in mass flow estimation because measurement magnitudes vary widely. For the nth measurement, let y_n be the true value, $\hat{y}_n$ be the predicted value, and $e_n = \hat{y}_n - y_n$ be the corresponding error. The RE is then computed on the original scale as

$$\mathrm{RE}_n = \frac{e_n}{y_n} \times 100\%. \tag{8}$$

They are the main evaluation metric in this study, with the 95th percentile taken as the primary basis for model comparison, often being the most crucial in industrial applications.

The *average normalized root-mean-squared error* (avgNRMSE), *range normalized root-mean-squared error* (rNRMSE), *mean absolute error* (MAE), *median absolute error* (MedAE), *mean absolute percentage error* (MAPE), *median absolute percentage error* (MedAPE), and R^2 are also used to obtain a comprehensive evaluation of the models:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} e_n^2}, \qquad R^2 = 1 - \frac{\sum_{n=1}^{N} e_n^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2},$$

$$\mathrm{rNRMSE} = \frac{\mathrm{RMSE}}{\max(\boldsymbol{y}) - \min(\boldsymbol{y})}, \qquad \mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N} |e_n|,$$

$$\mathrm{avgNRMSE} = \frac{\mathrm{RMSE}}{\bar{y}}, \qquad \mathrm{MedAE} = \mathrm{med}\big(|\boldsymbol{e}|\big),$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{n=1}^{N}\left|\frac{e_n}{y_n}\right|, \qquad \mathrm{MedAPE} = \mathrm{med}\left(\frac{|\boldsymbol{e}|}{\boldsymbol{y}}\right) \times 100\%$$

where $\boldsymbol{e} = (e_1, e_2, \dots, e_N)$, $\boldsymbol{y} = (y_1, y_2, \dots, y_N)$, $\bar{y}$ is the mean of y, and $\mathrm{med}(\cdot)$ denotes the median operator.

Normalized RMSE metrics are preferred over *mean squared error* (MSE) as it enables comparisons across datasets with different scales and can be normalized in different ways. The rNRMSE indicates how the average RMSE compares to the whole data range, whereas the avgNRMSE relates it to the mean of the data. Both metrics are included since the original data exhibit large variability, and we require the model to perform well in both scenarios.

IV. Implementation

An initial aliasing assessment showed that all features were at risk of aliasing, particularly the mass flow readings when the data were downsampled to 1 Hz or lower. Consequently, a low-pass FIR filter with a length of 129 samples and a cutoff frequency set to 80% of the Nyquist frequency was used to reduce aliasing before downsampling.

Each experiment was assigned a group number to identify different operating points, and the groups were defined by the chronological order of the experiments. In other words, the first experiment was assigned to group 1, the second to group 2, and so forth, yielding an "even" and "odd" split. Nine groups were excluded because they contained invalid or too few measurements. The final even split consisted of 148 780 measurements and 173 groups, whereas the odd split had 145 340 measurements and 169 groups.

Training and test splits were defined at the group level using this even/odd partition, and the stepwise experimental design ensured that both splits covered a wide range of all operating variables. To ensure representative and sound results, we performed two complementary runs: models were trained on the even split and tested on the odd split, and vice versa.

TABLE II
FINAL CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

Layer	Configuration
Input	$B \times W \times 5$ (five features, W timesteps)
Conv1D + ReLU	$5 \to 16$, kernel size = 3, padding = 1
Conv1D + ReLU	$16 \to 8$, kernel size = 3, padding = 1
Global AvgPool1D	Output: $B \times 8$
Fully connected + ReLU	$8 \to 16$
Fully connected (output)	$16 \to 1$

B: batch size. W: window length. Conv1D: one-dimensional convolution. ReLU: rectified linear unit activation function. AvgPool1D: one-dimensional average pooling.

TABLE III
HYPERPARAMETERS FOR THE FINAL MODELS

	Model	Window length	Batch size	Learning rate		Weight decay	
				Even	Odd	Even	Odd
4s	MLP	-	2	10^{-3}	10^{-3}	10^{-4}	10^{-4}
	MLPw	5	2	10^{-4}	10^{-4}	10^{-4}	10^{-5}
	CNN	5	2	10^{-4}	10^{-4}	10^{-4}	10^{-3}
	LSTM	6	2	10^{-4}	10^{-4}	10^{-4}	10^{-4}
60s	MLP	-	2	10^{-3}	10^{-3}	10^{-3}	10^{-5}
	MLPw	2	2	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	CNN	2	2	10^{-3}	10^{-3}	10^{-5}	10^{-5}
	LSTM	2	2	10^{-3}	10^{-5}	10^{-3}	10^{-5}

From the test split, one-third was set aside as a hold-out validation set to monitor early stopping when training the final model, and was not used in the final test evaluation. The validation and test sets thus comprised only operating points unseen during training and hyperparameter tuning, supporting model generalization to unseen flow conditions within the same experimental setup. Although training on approximately 50% of the data is unusual, it was deemed necessary to evaluate the model across a wide range of conditions. Splits were normalized (min–max) using the scaling parameters of the training set.

To ensure consistency and robustness of the results, all models were trained on 30 different seeds using the same splits and hyperparameters. Five features were used as inputs for the models: the temperature and apparent mass flow outputs from both CMFs in the skid, and the readings from the pressure transmitter. Models were trained at multiple downsampling intervals (250 ms, 500 ms, 1 s, 2 s, 3 s, 4 s, 5 s, and 6 s), and compared with models trained on the original data. A group-aware fivefold CV with a rolling window was used to tune the hyperparameters on the training set and avoid data leakage.

Models were trained using the MSE loss function, the *rectified linear unit* (ReLU) activation function, and the AdamW optimizer, for up to 60 epochs, with early stopping (patience = 10). Different network structures were explored; however, the final MLPs used one hidden layer, with 8 hidden nodes for the baseline and 16 hidden nodes for the MLPw. The final network structure of the CNN is shown in Table II. The final LSTM model consisted of a single recurrent layer with 10 hidden units and used the same input and output dimensions as the CNN. Initial window lengths were selected to capture approximately one dominant oscillation in the apparent mass flow signal from the liquid reference meter while retaining sufficient samples per experiment for training. They were later refined during hyperparameter tuning. Samples discarded during window initialization in the sequence models were likewise removed for the MLP, so that all models were trained and evaluated on identical data.

Implementation was done in PyTorch on a MacBook Air with an Apple M2 chip (8-core CPU) and 16 GB of memory.

V. RESULTS AND DISCUSSION

Downsampling the data yielded consistently better models across all metrics than training on the original sampling rate. Decreasing the sampling frequency improved the model performance, peaking at 4 s, although the 3 s and 5 s intervals yielded similar results. Only the results from models sampled at 4 s are discussed henceforth—denoted as "4 s"—as the other rates do not provide additional insights. The models trained on data averaged across entire experiments are denoted "60 s."

The tuned hyperparameters are listed in Table III. Sufficient variations were considered to confirm the robustness of the results, although further gains are likely with more extensive hyperparameter tuning, particularly of window lengths and network architectures.

A. Model Performance

1) Relative Errors: Table IV shows a collection of metrics related to the REs, computed over all 30 seeds rather than per-seed averages. For example, the "Max" value corresponds to the maximum error observed across all seeds. The CNN achieves the best overall performance and outlier robustness across both splits, while both the MLPw and LSTM substantially outperform the standard MLP. This consistent ranking indicates that incorporating temporal context improves CMF error correction beyond what is obtained with a nontemporal model. This is particularly clear when comparing the MLP and MLPw, which share the same underlying feedforward architecture and differ only in that MLPw receives short-term temporal input. The markedly better performance of MLPw therefore supports the conclusion that the gains arise from temporal information rather than increased model capacity.

Using shorter averages from each experiment yielded lower REs compared to taking one average across an entire experiment. Table IV demonstrates that the 4s models generally outperform the 60s models, especially for the sequence models. Only in the odd split do the 60s CNN and 60s LSTM have lower maximum errors than their 4s counterparts. However, relative to the original CMF outputs (Table IV, Max column), the maximum error decreases by 6 and 183% points for the 60s and 4s CNN models, respectively. Thus, the 4s CNN still performs well, particularly since occasional outliers are generally acceptable.

While all sequence models outperform the nontemporal MLP, the CNN achieves the strongest percentile and coverage metrics, suggesting that the most useful information is largely captured by local temporal patterns rather than longer-range dependencies. This interpretation is consistent with the CNN architecture: it preserves local temporal structure and uses

TABLE IV
RELATIVE ERROR METRICS (MAXIMA, PERCENTILES, AND COVERAGE WITHIN 10% AND 5%) COMPUTED FROM ALL ERRORS OVER 30 SEEDS (NO PER-SEED AVERAGING). BEST RESULTS IN BOLD; VALUES IN %

Split	Model		Max	p99	p95	p50	≤ 10%	≤ 5%
Even	MLP	4s	146	39.4	24.6	4.93	73.9	50.5
		60s	86.4	59.7	42.8	6.42	62.3	40.4
	MLPw	4s	53.5	26.1	16.2	3.58	86.3	63.8
		60s	205	62.4	25.1	5.31	70.3	48.1
	CNN	4s	44.4	**22.1**	**13.3**	**3.11**	**90.8**	**69.6**
		60s	158	41.8	16.1	5.27	80.4	47.7
	LSTM	4s	**41.0**	23.1	15.9	3.22	86.5	66.2
		60s	183	53.1	29.0	5.37	71.2	47.3
	Original	4s	168	84.5	47.1	8.57	54.5	38.1
		60s	69.4	57.1	48.5	9.50	52.0	31.4
Odd	MLP	4s	225	46.0	26.9	5.32	72.0	47.9
		60s	100	55.0	25.0	5.50	68.5	47.3
	MLPw	4s	63.6	34.6	17.0	3.58	85.0	64.2
		60s	65.6	45.7	28.2	4.93	69.9	50.7
	CNN	4s	55.8	**24.0**	**12.9**	**2.87**	**90.9**	**71.9**
		60s	**44.5**	25.7	17.8	5.17	75.1	48.2
	LSTM	4s	62.8	33.3	16.2	3.24	82.7	64.6
		60s	48.0	28.9	23.3	8.94	55.8	32.0
	Original	4s	239	78.9	51.8	6.83	58.3	43.1
		60s	50.6	46.6	41.9	10.0	49.5	30.5

TABLE V
RELATIVE ERROR METRICS BY GVF-RANGE, COMPUTED FROM ALL ERRORS OVER 30 SEEDS (NO PER-SEED AVERAGING) FOR THE 4S AND 60S CNN MODELS. BEST RESULTS IN BOLD; VALUES IN %

Split	Range	Model	n	Max	p99	p95	p50	≤ 10%	≤ 5%
Even	0–15 %	4s	13,140	**27.2**	**12.8**	**8.35**	**2.40**	**97.2**	**82.5**
		60s	990	158	146	32.8	4.70	86.1	57.7
	15–35 %	4s	9,450	25.0	18.7	**13.6**	**3.21**	**91.4**	**69.3**
		60s	1050	**22.0**	**16.3**	14.0	5.28	83.9	47.0
	35–50 %	4s	2,850	29.8	25.8	22.5	**3.74**	**83.6**	**63.4**
		60s	300	**21.0**	**18.6**	**15.2**	5.88	79.3	42.3
	50–70 %	4s	5,580	31.0	**17.7**	**12.8**	**4.28**	**88.1**	**56.3**
		60s	420	**25.6**	21.6	16.4	7.13	68.8	36.7
	70–80 %	4s	1,230	44.4	34.1	25.5	8.41	58.5	23.9
		60s	120	**24.8**	**22.8**	**16.5**	**5.81**	**74.2**	**40.8**
	80–95 %	4s	1,230	43.4	31.9	**18.9**	**4.63**	**79.3**	**53.6**
		60s	180	**36.8**	**31.7**	23.9	7.63	61.7	36.1
Odd	0–15 %	4s	14,820	**24.7**	**12.9**	**7.91**	**2.24**	**97.6**	**82.6**
		60s	930	25.8	21.6	14.6	4.40	87.9	57.4
	15–35 %	4s	9,030	**23.0**	**16.4**	**13.3**	**3.02**	**89.2**	**70.6**
		60s	840	42.5	23.4	17.5	4.96	74.5	50.4
	35–50 %	4s	3,900	**20.3**	**14.0**	**11.3**	**3.80**	**92.1**	**63.4**
		60s	360	40.0	32.8	21.9	5.71	68.1	45.8
	50–70 %	4s	3,270	**30.9**	**17.0**	**13.5**	**3.11**	**88.8**	**68.7**
		60s	450	44.5	33.6	18.4	8.10	61.1	31.3
	70–80 %	4s	1,020	27.3	21.9	**15.7**	**7.7**	**69.8**	25.0
		60s	150	**22.3**	**20.4**	17.4	9.4	52.0	**30.0**
	80–95 %	4s	1,980	55.8	43.9	37.8	6.34	61.1	43.9
		60s	120	**19.5**	**17.2**	**15.0**	**4.15**	**82.5**	**55.0**

Here n is the total number of samples (test points multiplied by 30 seeds).

shared convolutional filters across the input window, allowing local patterns to be detected at different time positions in the sequence. The best model (4s CNN) shows a relative reduction of about 77% compared to the largest original RE, demonstrating both superior performance and a substantial reduction of the original CMF errors, as is illustrated in Fig. 4 for the odd data split. The right subfigure includes all REs across 30 seeds, whereas the left subfigure shows the original REs for the same test points. Approximately 95% of the REs are below 13% in estimating the overall mass flow rate, and the majority are less than 3%. Results on the even split are highly similar.

Fig. 4 shows that REs increase with GVF, consistent with prior work. To quantify this trend, Table V reports the same RE metrics as Table IV but separated into six GVF ranges for the best 4s and 60s models over the 30 seeds. As GVF increases, the error distribution worsens: the 95th percentile rises and the proportions of REs within 10% and 5% decline, with a few exceptions at the highest GVFs. The 4s CNN generally outperforms the 60s CNN—lower 95th percentiles in 9/12 ranges, lower medians in 10/12, and higher ≤10% coverage in 10/12—while the 60s CNN performs better only in a few high-GVF ranges. The 4s split contains about 8 to 14 times more samples and exhibits higher unprocessed errors; its empirical 99th percentile and maximum may therefore be larger simply because of greater exposure to rare extremes. We thus emphasize the median, the 95th percentile, and coverage when assessing performance.

2) Complementary Metrics: All evaluation metrics except the RE-based metrics are reported in Table VI for the 4s and

TABLE VI
AVERAGE METRICS FOR THE BEST 4S AND 60S MODELS (CNN) ACROSS SEEDS ± STD. DEV. BEST RESULTS IN BOLD

Split	Metric	CNN 4s		CNN 60s		Original data 4s	Original data 60s
Even	rNRMSE	**0.03**	**± 0.002**	0.05	± 0.004	0.08	0.08
	avgNRMSE	**0.05**	**± 0.004**	0.09	± 0.008	0.14	0.14
	MAE [$\mathrm{kg\,h^{-1}}$]	**265**	**± 20.7**	440	± 45.06	700	736
	MedAE [$\mathrm{kg\,h^{-1}}$]	**183**	**± 21.6**	304	± 46.28	462	506
	MAPE [%]	**4.37**	**± 0.58**	7.70	± 0.84	15.5	16.4
	MedAPE [%]	**3.14**	**± 0.41**	5.37	± 0.99	8.61	9.50
	R^2	**0.99**	**± 0.001**	0.98	± 0.004	0.94	0.94
Odd	rNRMSE	**0.03**	**± 0.002**	0.04	± 0.007	0.07	0.07
	avgNRMSE	**0.05**	**± 0.003**	0.09	± 0.014	0.12	0.14
	MAE [$\mathrm{kg\,h^{-1}}$]	**246**	**± 22.5**	409	± 86.05	599	674
	MedAE [$\mathrm{kg\,h^{-1}}$]	**179**	**± 28.1**	317	± 95.7	397	409
	MAPE [%]	**4.25**	**± 0.58**	7.00	± 1.58	14.6	14.7
	MedAPE [%]	**2.90**	**± 0.45**	5.55	± 1.42	7.00	10.0
	R^2	**0.99**	**± 0.001**	0.98	± 0.007	0.96	0.96

60s CNN models, together with the corresponding baseline metrics obtained by using the apparent CMF outputs as predictions $\hat{y}$. The NRMSE values and R^2 indicate high predictive performance when considered in isolation. However, these metrics should be interpreted with care, since the baseline values are already high (e.g., $R^2 = 0.94$). These metrics are therefore not ideal for assessing model performance in isolation, as they do not fully reflect the difficulty of the correction task; however, they still show that the models are stable

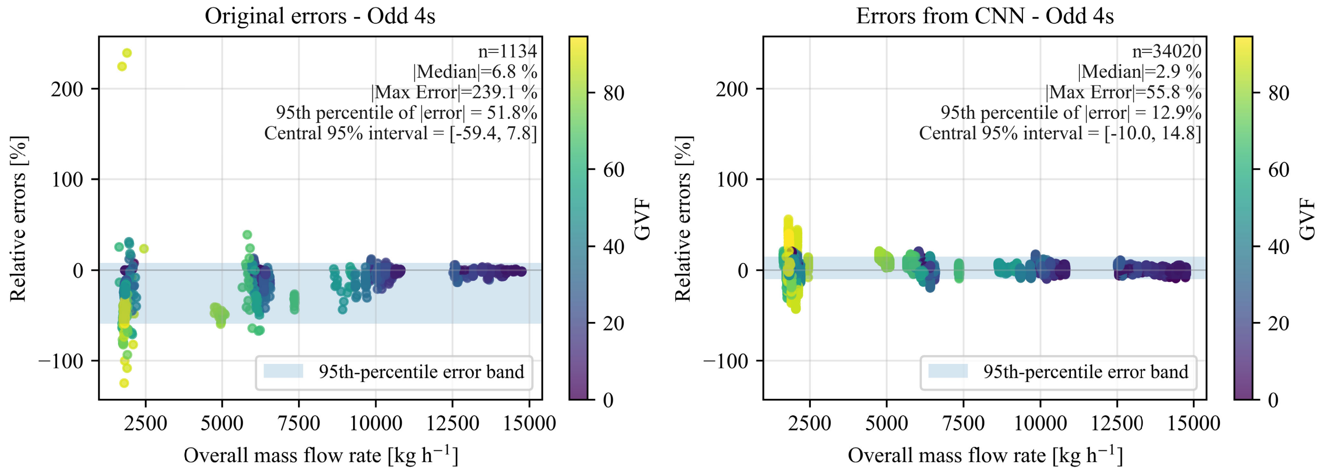


Fig. 4. REs versus overall mass flow rate for the best odd model, compared with the original errors on the same test points.

and improve on the baseline. The MedAE is substantially lower than the MAE, indicating that the error distribution is influenced by a subset of larger errors. MedAE may therefore better reflect the typical model performance, especially if a few outliers are acceptable. The percentile metrics MAPE and MedAPE are both greatly reduced, particularly for the 4s CNN, suggesting that this model is more accurate, robust, and consistent across a variety of conditions. Overall, Table VI shows that several complementary metrics are needed to fully evaluate model performance, and that short-time averaging is beneficial.

B. Robustness Analysis

Models trained on the odd split generally performed slightly better than those on the even split, possibly because of the larger number of samples in the 0%–15% GVF range. Nonetheless, the results are similar, as shown in Tables IV–VI. All previous comments regarding the benefits of sequence models and short-time averages hold for both splits. Despite the group-parity variability, especially evident in Table V, the overall results are consistent across splits, indicating that the findings are not an artifact of a favorable split.

Fig. 5 shows, for the 4s and 60s CNNs, the average percentage of REs (across seeds) that fall within a given percentile threshold. The 4s CNN has only small percentage-point differences between the splits, while the between-seed variability is minimal and nearly identical across splits, as indicated by the overlapping shaded areas in the top subfigure. The 60s CNN exhibits greater between-seed and group-parity variability, suggesting that it is not as robust as the 4s CNN. Furthermore, the standard deviations in Table VI indicate limited variability for the 4s CNN, further supporting its consistency. Together, these results confirm that the 4s CNN delivers robust and consistent performance across both data splits and random seeds.

C. Model Complexity

Table VII shows different metrics used to assess the complexity of all four models when trained on the 4s downsampled data. Latency is often the most important metric in real-time applications as they require near-instant responses, and the

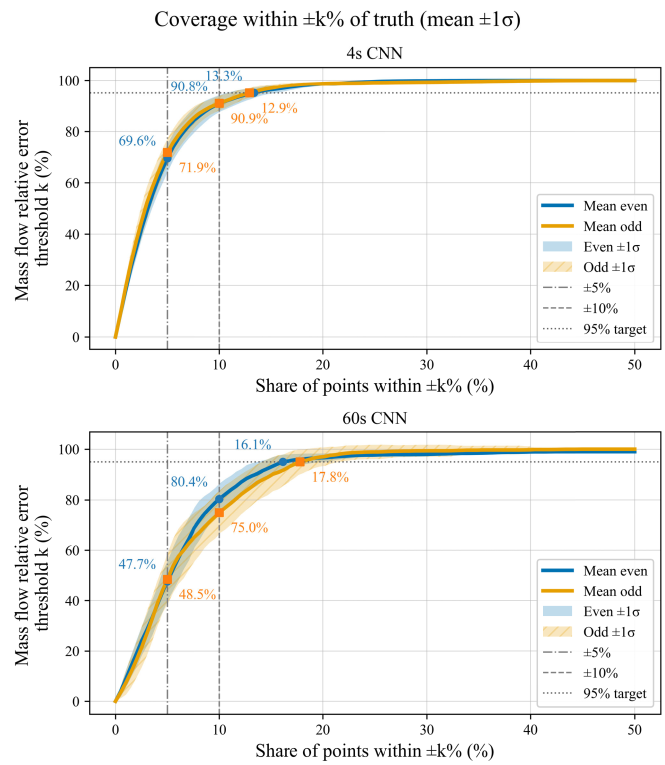


Fig. 5. Coverage percentiles averaged per seed for both data splits. Results are shown for the 4s CNN (top) and 60s CNN (bottom) models. Shading indicates ±1 std. dev. across seeds.

TABLE VII
COMPLEXITY METRICS FOR MODELS TRAINED ON THE 4S DATA. LATENCY REFERS TO THE MEAN ACROSS SEEDS ± STD. DEV

Model	Parameters	MACs per example	Latency [µs per example]	
			Even	Odd
MLP	57	96	30 ± 1.3	31 ± 1.3
MLPw	433	832	27 ± 2.6	28 ± 3.1
CNN	809	6,528	170 ± 46	160 ± 39
LSTM	925	8,184	172 ± 11	163 ± 13

MLPw can thus be a solid alternative to the CNN if latency is prioritized over a small loss in accuracy. The MLPw was consistently the fastest of the four models for all downsampling intervals, despite having more parameters and *multiply–accumulate operations* (MACs) than the MLP. The two MLPs only differed by 3 μs, and both were roughly 6× faster than the CNN and LSTM, whose latencies were in the range of 160–172 μs. By parameter count and MACs, the LSTM was the most complex model, followed by the CNN. However, all models were relatively fast and simple compared to many modern ML architectures.

VI. Conclusion and Future Work

This study demonstrated that correcting CMF measurements by exploiting temporal dependencies and applying short-time averaging within experiments yields significantly improved results in terms of lower REs, MAPEs, NRMSEs, and R^2 values. Four models (an MLP, an MLPw, an LSTM, and a 1D-CNN) were compared across various downsampling intervals. The best performance was obtained by the CNN sampled at 0.25 Hz, achieving about 95% of REs below 13%, an rNRMSE of 0.03, and a MAPE of about 4.3%; however, all sequence models significantly outperformed the nontemporal baseline. The models were trained on three-phase air–water–oil flow data covering a wide range of operating conditions, and achieved strong performance on a test set consisting of operating conditions not observed during training, indicating good generalization within the same experimental setup. Since all measurements were acquired using the same instruments and test facility, model generalization to other setups is not established and would require more research. Nevertheless, we strongly expect temporal information and short-time averaging to remain important when developing correction models for different experimental setups. Robustness was assessed by training and evaluating on two distinct splits and 30 random seeds. Further research may explore different network structures and models, as well as time-aware feature engineering to improve the accuracy, especially for higher GVFs.

References

[1] C. Mills, G. Chinello, and M. Henry, "Flow measurement challenges for carbon capture, utilisation and storage," *Flow Meas. Instrum.*, vol. 88, Dec. 2022, Art. no. 102261.

[2] A. Bista, S. A. Hogan, C. P. O'Donnell, J. T. Tobin, and N. O'Shea, "Evaluation and validation of an inline Coriolis flowmeter to measure dynamic viscosity during laboratory and pilot-scale food processing," *Innov. Food Sci. Emerg. Technol.*, vol. 54, pp. 211–218, Jun. 2019.

[3] M. M. F. Figueiredo, J. L. Goncalves, A. M. V. Nakashima, A. M. F. Fileti, and R. D. M. Carvalho, "The use of an ultrasonic technique and neural networks for identification of the flow pattern and measurement of the gas volume fraction in multiphase flows," *Experim. Thermal Fluid Sci.*, vol. 70, pp. 29–50, Jan. 2016.

[4] L. Xu, W. Zhou, X. Li, and S. Tang, "Wet gas metering using a revised Venturi meter and soft-computing approximation techniques," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 3, pp. 947–956, Mar. 2011.

[5] T. Wang and R. Baker, "Coriolis flowmeters: A review of developments over the past 20 years, and an assessment of the state of the art and likely future directions," *Flow Meas. Instrum.*, vol. 40, pp. 99–123, Dec. 2014.

[6] Y. Arellano, S.-A. Tjugum, O. B. Pedersen, M. Breivik, E. Jukes, and M. Marstein, "Measurement technologies for pipeline transport of carbon dioxide-rich mixtures for CCS," *Flow Meas. Instrum.*, vol. 95, Feb. 2024, Art. no. 102515.

[7] R. P. Liu, M. J. Fuent, M. P. Henry, and M. D. Duta, "A neural network to correct mass flow errors caused by two-phase flow in a digital Coriolis mass flowmeter," *Flow Meas. Instrum.*, vol. 12, no. 1, pp. 53–63, Mar. 2001.

[8] J. A. Weinstein, "Multiphase flow in Coriolis mass flow meters–error sources and best practices," in *Proc. 28th Int. North Sea Flow Meas. Workshop*, 2010, p. 17. [Online]. Available: https://nfogm.no/documents/north-sea-flow-measurement-workshop/nsfmw-2010/

[9] M. Tombs, M. Henry, H. Yeung, and R. Lansangan, "Coriolis mass flow meter developments: Increasing the range of applications in oil & gas production and processing," in *Proc. North Sea Flow Meas. Workshop*, 2004, p. 20. [Online]. Available: https://nfogm.no/documents/north-sea-flow-measurement-workshop/nfsmw-2004/

[10] L. S. Hansen, S. Pedersen, and P. Durdevic, "Multi-phase flow metering in offshore oil and gas transportation pipelines: Trends and perspectives," *Sensors*, vol. 19, no. 9, p. 2184, May 2019.

[11] Y. Yan, L. Wang, T. Wang, X. Wang, Y. Hu, and Q. Duan, "Application of soft computing techniques to multiphase flow measurement: A review," *Flow Meas. Instrum.*, vol. 60, pp. 30–43, Apr. 2018.

[12] S. Bahrami, S. Alamdari, M. Farajmashaei, M. Behbahani, S. Jamshidi, and B. Bahrami, "Application of artificial neural network to multiphase flow metering: A review," *Flow Meas. Instrum.*, vol. 97, Jul. 2024, Art. no. 102601.

[13] L. Wang, J. Liu, Y. Yan, X. Wang, and T. Wang, "Gas-liquid two-phase flow measurement using Coriolis flowmeters incorporating neural networks," in *IEEE Int. Instrum. Meas. Technol. Conf. Proc.*, May 2016, pp. 1–5.

[14] L. Wang, J. Liu, Y. Yan, X. Wang, and T. Wang, "Mass flow measurement of two-phase carbon dioxide using coriolis flowmeters," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2017, pp. 1–5.

[15] L. Wang, Y. Yan, X. Wang, T. Wang, Q. Duan, and W. Zhang, "Mass flow measurement of gas-liquid two-phase CO_2 in CCS transportation pipelines using Coriolis flowmeters," *Int. J. Greenhouse Gas Control*, vol. 68, pp. 269–275, Jan. 2018.

[16] C. Sun, Y. Yan, W. Zhang, and L. Wang, "A dynamic ensemble selection approach to developing softcomputing models for two-phase flow metering," *J. Phys., Conf. Ser.*, vol. 1065, Aug. 2018, Art. no. 092022.

[17] C. Sun, L. Wang, Y. Yan, W. Zhang, and D. Shao, "A novel heterogeneous ensemble approach to variable selection for gas-liquid two-phase CO_2 flow metering," *Int. J. Greenhouse Gas Control*, vol. 110, Sep. 2021, Art. no. 103418.

[18] Q. Wang, Y. Yan, W. Zhang, Y. Zhang, and H. Li, "Performance evaluation of a Coriolis flowmeter for measuring mass flow rate of CO_2 with impurities under CCS conditions," *Meas., Sensors*, vol. 38, p. 101558, May 2024.

[19] W. S. Chowdhury, Y. Yan, M.-A. Coster-Chevalier, and J. Liu, "Mass flowrate measurement of slurry using coriolis flowmeters and data driven modeling," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.

[20] M. Henry, M. Tombs, M. Zamora, and F. Zhou, "Coriolis mass flow metering for three-phase flow: A case study," *Flow Meas. Instrum.*, vol. 30, pp. 112–122, Apr. 2013.

[21] M. Bao, M. Wang, K. Li, and X. Jia, "Integrating machine learning with sensor technology for multiphase flow measurement: A review," *IEEE Sensors J.*, vol. 24, no. 19, pp. 29603–29618, Oct. 2024.

[22] T. Bikmukhametov and J. Jäschke, "First principles and machine learning virtual flow metering: A literature review," *J. Petroleum Sci. Eng.*, vol. 184, Jan. 2020, Art. no. 106487.

[23] H. Wang, M. Zhang, and Y. Yang, "Machine learning for multiphase flowrate estimation with time series sensing data," *Measurement: Sensors*, vols. 10–12, Nov. 2020, Art. no. 100025.

[24] H. Wang, D. Hu, M. Zhang, and Y. Yang, "Multiphase flowrate measurement with time series sensing data and sequential model," *Int. J. Multiphase Flow*, vol. 146, Jan. 2022, Art. no. 103875.

[25] H. Wang, D. Hu, M. Zhang, N. Li, and Y. Yang, "Multiphase flowrate measurement with multimodal sensors and temporal convolutional network," *IEEE Sensors J.*, vol. 23, no. 5, pp. 4508–4517, Mar. 2023.

[26] Y. Jiang et al., "A flow rate estimation method for gas–liquid two-phase flow based on transformer neural network," *IEEE Sensors J.*, vol. 24, no. 16, pp. 26902–26913, Aug. 2024.

[27] M. Bao, R. Wu, M. Wang, K. Li, and X. Jia, "Enhancing accuracy in gas–water two-phase flow sensor systems through deep-learning-based computational framework," *IEEE Sensors J.*, vol. 24, no. 23, pp. 39934–39946, Jun. 2024.

[28] Z. Gao et al., "Multitask-based temporal-channelwise CNN for parameter prediction of two-phase flows," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6329–6336, Sep. 2021.

[29] L. Zhang, Y. Liu, and J. Liu, "Gas volume fraction measurement for gas-liquid two-phase flow based on dual CNN-transformer mixture neural network," *IEEE Sensors J.*, vol. 25, no. 13, pp. 25108–25118, Jul. 2025.

[30] Y. Zhang, Y. Liu, Z. Liu, and W. Liang, "Developing a long short-term memory-based signal processing method for Coriolis mass flowmeter," *Measurement*, vol. 148, Dec. 2019, Art. no. 106896.

[31] L. Sun, Z. Zhao, H. Wang, and X. Zhuang, "Coriolis flowmeter for two-phase flow measurement based on TCN-LSTM," in *Proc. 43rd Chin. Control Conf. (CCC)*, Jul. 2024, pp. 3452–3458.

[32] R. Thorn, G. A. Johansen, and B. T. Hjertaker, "Three-phase flow measurement in the petroleum industry," *Meas. Sci. Technol.*, vol. 24, no. 1, Jan. 2013, Art. no. 012003.

[33] Krohne. (2025). *OPTIMASS 6000 Coriolis Mass Flowmeter*. [Online]. Available: https://cdn.krohne.com/pick2/tagged_docs/TD_OPTIMASS_6000_en_250122_4001894106_R06.pdf

[34] KROHNE. (2023). *OPTIBAR PM 3050 Pressure Transmitter*. [Online]. Available: https://cdn.krohne.com/pick2/tagged_docs/TD_OPTIBAR-PM-3050_en_230124_4007693201_R01.pdf

[35] S. J. Prince, *Understanding Deep Learning*. Cambridge, MA, USA: MIT Press, 2023. [Online]. Available: http://udlbook.com

[36] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[38] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

Amanda Nyholm received the B.Sc. degree in mathematics and the M.Sc. degree in mathematical statistics from Lund University, Lund, Sweden, in 2020 and 2022, respectively. She is currently pursuing the Ph.D. degree with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, within the Signal Processing Research Group.

Her research interests include multivariate time series, signal processing, machine learning, and statistics.

Yessica Arellano received the B.E. degree from La Universidad del Zulia, Maracaibo, Venezuela, in 2005, the M.Sc. (Hons.) degree from Robert Gordon University, Aberdeen, U.K., in 2007, the second M.Sc. degree from La Universidad del Zulia, in 2011, and the Ph.D. degree from Coventry University, Coventry, U.K., in 2020, focused on advanced multiphase flow monitoring via electromagnetic measurements.

She is currently a Senior Research Scientist with the Department of Gas Technology, SINTEF Energy Research, Trondheim, Norway, specializing in flow measurement technologies for carbon dioxide transport systems. She is also the inventor behind two patented innovations for apparatuses and methods for measuring multiphase flow.

Jinyu Liu received the joint B.Eng. degree from Tianjin University, Tianjin, China, and the University of Kent, Canterbury, U.K., in 2012, the M.Sc. degree from the University of Southampton, Southampton, U.K., in 2013, and the Ph.D. degree in electronic engineering from the University of Kent in 2019.

He is currently a Research Engineer at the Research and Development Department, KROHNE Ltd., Wellingborough, U.K. His research interests include Coriolis flowmeter technology, multiphase flow measurement, condition monitoring of industrial processes, and digital signal processing.

Damian Krakowiak received the B.Sc. (Hons.) degree in electrical and electronic engineering from Glasgow Caledonian University, Glasgow, U.K., in 2012.

He is currently the Senior Metrology Manager and a Custody Transfer Approvals Engineer at KROHNE Ltd., Wellingborough, U.K., specializing in metrology, fiscal measurement, and ISO 17025 compliance. With prior technical leadership at TÜV SÜD, he has extensive experience in flow measurement, calibration automation, and quality system development. His work focuses on advancing metrological standards and ensuring precision within complex fiscal and custody transfer applications.

Pierluigi Salvo Rossi (Senior Member, IEEE) was born in Naples, Italy, in 1977. He received the Dr.Eng. (summa cum laude) degree in telecommunications engineering and the Ph.D. degree in computer engineering from the University of Naples Federico II, Naples, in 2002 and 2005, respectively.

He worked with the University of Naples Federico II, the Second University of Naples, Naples, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, and Kongsberg Digital AS, Asker, Norway. He held visiting appointments with Drexel University, Philadelphia, PA, USA; Lund University, Lund, Sweden; NTNU; and Uppsala University, Uppsala, Sweden. He is currently a Full Professor and the Deputy Head of the Department of Electronic Systems, NTNU. He is also a part-time Senior Research Scientist with the Department of Gas Technology, SINTEF Energy Research, Trondheim, Norway. His research interests fall within the areas of communication theory, data fusion, machine learning, and signal processing.

Prof. Salvo Rossi was awarded as an Exemplary Senior Editor of IEEE COMMUNICATIONS LETTERS in 2018. He is (or has been) on the Editorial Board of IEEE SENSORS JOURNAL, IEEE OPEN JOURNAL OF COMMUNICATIONS SOCIETY, IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, IEEE COMMUNICATIONS LETTERS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.