Digital Chemical Engineering 2 (2022) 100012

Contents lists available at ScienceDirect

Digital Chemical Engineering

journal homepage: www.elsevier.com/locate/dche

Gas-liquid flow regimes identification using non-intrusive Doppler ultrasonic sensor and convolutional recurrent neural networks in an s-shaped riser

Boyu Kuang[a,c,1], Somtochukwu Godfrey Nnabuife[b,1], Shuang Sun[c,*], James F. Whidborne[d,*], Zeeshan A. Rana[a]

[a] Centre for Computational Engineering Sciences, Cranfield University, Cranfield MK43 0AL, United Kingdom
[b] Geo-Energy Engineering Centre, Cranfield University, Cranfield MK43 0AL, United Kingdom
[c] Tianjin Key Laboratory of Airworthiness and Maintenance of Civil Aircraft, Department of Aviation Engineering, Civil Aviation University of China, 2898 Jinbei Road, Dongli District, Tianjin 300300, China
[d] Dynamics Simulation and Control Group, Cranfield University, Cranfield MK43 0AL, United Kingdom

ARTICLE INFO

Keywords:
Two-phase flow
Flow regime identification
Ultrasonic signal
Time-domain property
Deep learning

ABSTRACT

The problem of gas-liquid (two-phase) flow regime identification in an S-shaped riser using an ultrasonic sensor and convolutional recurrent neural networks (CRNN) is addressed. This research systematically evaluates three different schemes with four CRNN-based classifiers over fourteen experiments. Four metrics are used as the evaluation criteria: categorical accuracy, categorical cross-entropy, mean square error (MSE), and computation graph complexity. Compared with existing results, a compatible performance is achieved while considerably reducing the model complexity. The testing and validation accuracies were 98.13% and 98.06%, while the complexity decreased by 98.4% (only 117,702 parameters). The proposed approach is i) accurate, low complexity, and non-intrusive and hence suitable for industry, and ii) could provide a benchmark for flow regime identification.

1. Introduction

The simultaneous flow of two-phase gas-liquid flow occurs in a wide range of processes in nature and technological applications (Shen and Hibiki, 2021), for example, petrochemical processes (Feng et al., 2021), chemical analysis and synthesis (Kou and Sun, 2018), and spraying process (Hammad et al., 2021). The knowledge and the understanding of the systems flow conditions are essential due to their impact on process intensification, safety issues, and process control in industrial applications (Nnabuife et al., 2021; Wiedemann et al., 2019). Because the central focus of Falcone et al. (2018) and Liu et al. (2018) is on flow recognition, intermittent or irregular flow structures are generally unwanted as they can cause plant dynamic mechanical stresses, or plant shut down (Nnabuife et al., 2019a). As a result, the determination and prediction of the spatial-temporal phase distribution is a necessity for safe and efficient plant operation (Kuang et al., 2021; Nnabuife and Pilario, 2019).

Different methodologies for identifying flow regimes have been proposed (Chakraborty and Das, 2018; Figueiredo et al., 2020; Xu, Li, et al., 2020). Rosa et al. (2010) published a study of the critical methods that have been widely investigated. Other approaches include optical imaging (Sunde et al., 2005), electrical impedance (George et al., 2000), pressure fluctuation (Chalgeri and Jeong, 2019), radiation absorption (Hanus et al., 2018), and ultrasound. The optical imaging system is non-invasive, low-cost, and capable of simultaneously providing information on several flow characteristics, although it needs a transparent pipe and fluids for optical measurements (Sunde et al., 2005). The pressure fluctuation method is also inexpensive and simple to use, while it necessitates sealing tape on the tubing, which may increase the risk of leakage (Chalgeri and Jeong, 2019). Electrical impedance is also a low-cost and simple-to-use procedure (George et al., 2000). On the other hand, impedance methods are susceptible to phase inversion in water-oil flows. They may also be particularly sensitive to the flow pattern within the channel. Radiation absorption is another method for identifying flow regimes. The absorption of radiation is a well-studied and non-invasive technique (Affonso et al., 2020; Hanus, 2015). The main drawback of the radiation absorption technique is its safety concerns and high cost.

Ultrasonic techniques are a promising alternative to radioactive techniques, which are expensive, complex, and unsafe. Ultrasonic

* Corresponding authors.
E-mail address: j.f.whidborne@cranfield.ac.uk (J.F. Whidborne).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.dche.2022.100012
Received 18 August 2021; Received in revised form 21 December 2021; Accepted 9 January 2022
2772-5081/© 2022 The Author(s). Published by Elsevier Ltd on behalf of Institution of Chemical Engineers (IChemE). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Nomenclature

$2f_s$	Source frequency
f_D	Doppler frequency shift
f_R	Frequency received
f_0	Frequency transmitted
L_b	Length of Taylor bubbles
L_s	Length of slug body
V_{GS}	Gas bubble velocity
V_{LS}	The velocity of liquid in slug
x_t	Signal transmitted
x_r	Signals received
w_s	Reference signal
x_D	Doppler signal
z	Acoustic impedance
θ_1	Phase term based on the shift in phase produced within the receiver and scatterer distance from the transducer

techniques are given due consideration because they are non-intrusive, non-invasive, fast, low-cost, and simple to operate, and suitable for real-time measurements. They can work in extreme environments such as high pressure and high temperatures. It also has high precision, is time sensitive, and is suitable for optically opaque systems (Figueiredo et al., 2020). A comprehensive review of the use of the ultrasonic technique in the context of multiphase flow measurement can be found in Tan et al. (2021). Ultrasonic signals can be used to meet the needs of long-distance signal transmission, remote sensing, and harsh operating conditions that are common in the oil industry. They have also been widely used to measure phase velocity and phase fraction (Thorn et al., 2013). For velocity measurement, the pulsed-wave ultrasonic Doppler (PWUD) and continuous-wave ultrasonic Doppler (CWUD) techniques have been investigated (Tan et al., 2021). PWUD intercepts the moving stream with a single transducer by sending short ultrasonic bursts and receiving echoes from tracer particles along a sound beam. Typical examples are Murai et al. (2010) and Yin et al. (2020). To determine the average flow velocity, the CWUD employs two separate transmitting and receiving transducers (Dong et al., 2015; Nnabuife et al., 2020; Nnabuife et al., 2019). Although PWDU measurements are perfectly adequate for flow measurement, the maximum flow velocity that can be measured with the PWDU is limited by its sampling frequency. The CWDU technique accurately measures two-phase flow velocity and has no maximum limit on the flow velocity it can measure (Tan et al., 2021). A lot of research work has been carried out using ultrasonic sensors for two-phase flow regime classification, and they can be found in Fang et al. (2020), Figueiredo et al. (2016), Nnabuife et al. (2021), Wada et al. (2006), and Mao et al. (2022).

Recent developments in machine learning and artificial intelligence have brought innovation to the signal processing process in two-phase flow identification (Lin et al., 2020; Parrales et al., 2018; Queiroz et al., 2021). More specifically, Nnabuife et al. (2019) proposed a baseline using the principal component analysis (PCA) and support vector machine (SVM), achieving 85% identification accuracy. The identification accuracy is relatively low because of the limited training data (Nnabuife et al., 2019). Subsequently, Nnabuife et al. (2020) proposed the twin-window feature extraction (TFE) algorithm to artificially augment the training dataset, which pushes the identification accuracy to 96.28% with a proposed deep neural network (DNN)-based classifier. Although Nnabuife et al. (2020) achieve a high-standard identification accuracy, its classifier requires a huge input data size (more than 50k data points), and the model size is also large. The full-connected structures (Nnabuife et al., 2020) have significantly low learning efficiency especially with a large amount of data. These drawbacks may make it difficult to implement this flow regime identification technology in real-world scenarios due to limited computational resources. Instead of tracing higher identification accuracy, Kuang et al. (2020) propose the pseudo-image-feature (PIF) algorithm to assist the explain-ability of the ultrasonic signal processing procedure, which extends the machine vision inspirations into the multiphase flow identification sector. However, the high model complexity (along with high computation resource requirement) exists (Kuang et al., 2020).

Their related researches (Kuang et al., 2020; Nnabuife et al., 2020; Nnabuife et al., 2019) are not end-to-end (E2E) solutions. The E2E refers to the functionality of inputting raw ultrasonic signals then outputting the identified flow regime (Saltzer et al., 1984). The existing solutions with multiple steps inevitably encounter the difficulty of optimization. Various steps correspond to various functionalities, which are not all orientating to flow regime identification. This E2E concentrates the entire solution to one clear and direct target, which improves efficiency. Furthermore, the proposed E2E model can be used as the pre-trained model for further transfer learning to fit other similar problems.

The convolutional recurrent neural network (CRNN) is a hybrid structure consisting of a convolutional neural network (CNN) and a recurrent neural network (RNN) (Deng et al., 2020; Zhang et al., 2021). Ultrasonic signals have characteristics of both low signal-noise ratio and long time-domain span (Nnabuife et al., 2019a). The parameter sharing property of CNN provides a feature extraction approach with low overfitting risk, which can enforce the robustness for the low signal-noise ratio (Zhai et al., 2016). However, the receptive field increase in CNN inevitably brings in high costs in both information loss and the risk of gradient vanishing (O'Shea and Nash, 2015). Notably, the common approaches are the Atrous (Dilated) Convolution, pooling, and stride-convolution layer to increase the receptive field (O'Shea and Nash, 2015). However, an RNN focuses on handling the history dependency (Ghavamian and Simone, 2019). An RNN iterates the numerical correlation among timestamps through a series structure and implicitly hides inside the RNN structure (He et al., 2020). The RNN has a relatively simple structure, but a long time-domain span might suppress the relation from the early timestamp. Therefore, combining the CNN feature extraction and RNN time-domain correlation becomes a very tempting idea, which is especially suitable for the low signal-noise ratio and long time-domain span ultrasonic signals in this research (Pascanu et al., 2013). The CRNN architecture has achieved promising effectiveness in other sectors (Jeon and Moon, 2020; Qin et al., 2019).

Flow regime identification in flexible risers has received more attention in recent years. Li et al. (2013) conducted experiments using air-water on three separate riser configurations, including a free-hanging catenary, a lazy S-riser, and a steep S-riser. It was discovered that slug flow in flexible risers behaved differently to classical flow in a vertical riser (Li et al., 2013). Some published research on flow regime identification in flexible risers can be found in Li et al. (2017), Xu et al. (2020) and Xu et al. (2020). Despite comprehensive attempts to research flow patterns in pipeline-riser systems, comparatively few studies on two-phase flow patterns in pipeline-riser systems with an S-shaped riser have been published. Furthermore, to the best of the authors' understanding, there is no documented work on identifying two-phase gas-liquid flow regimes on an S-shaped riser using an ultrasonic sensor and CRNN. The flow regime recognition in an S-shaped riser has not yet been extensively analyzed. Exploration and exploitation of offshore petroleum deposits can move into shallow waters while the majority of new oil and gas discoveries are projected to occur offshore. As a result, the offshore petroleum industry will see a greater use of floating processing facilities and flexible risers. The characteristics and mechanisms of two-phase gas-liquid flow in an S-shaped riser are critical not only for efficient pipe sizing, design, and routing, but also for the operation of the downstream system.

This research proposes an identification framework for the flow regime ultrasonic signals in the S-shaped riser based on the proposed CRNN model. This research aims to find the best E2E configuration for flow regime identification by fusing the CNN feature extraction and the RNN time-series analysis. Compared with the existing models, the goal

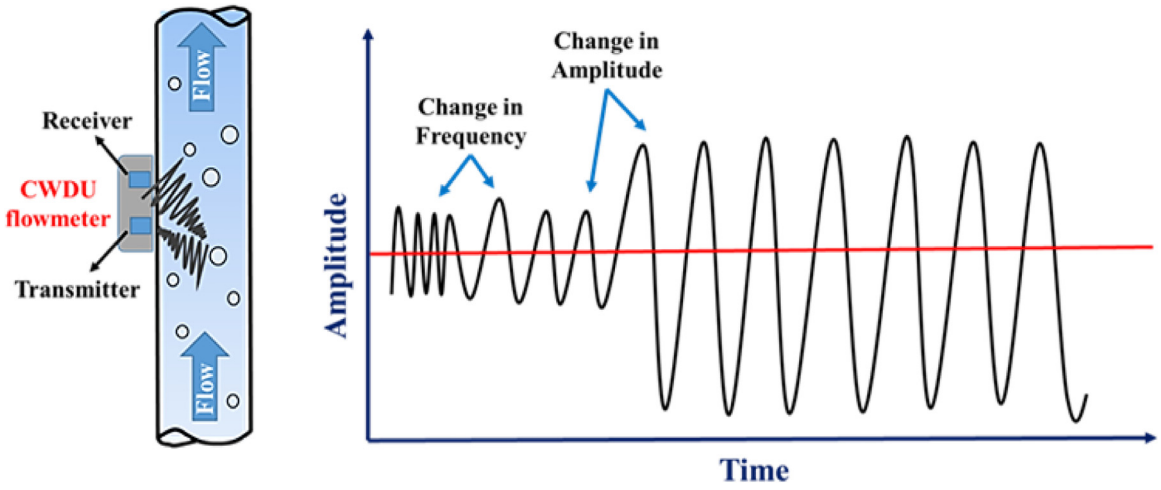


Fig. 1. The process of the Continuous Wave Doppler Ultrasound (CWDU) flowmeter (Nnabuife et al., 2020).

of the CRNN model proposed in this research is to significantly reduce the complexity of the model while ensuring a relatively high flow regime identification accuracy.

In summary, this research's main contributions are as follows: (i) this appears to be the first end-to-end (E2E) solution towards the gas-liquid flow regime identification using ultrasonic signals in an S-shaped riser. (ii) the proposed CRNN classifier significantly decreases the module complexity along with a high identification accuracy. (iii) to the best of the authors' knowledge, this research is the first successful trial of cooperating the feature extraction of CNN and time-domine of RNN for the flow regime identification. (iv) this paper investigated various CRNN schemes with systematic details, which can be a benchmark for further research.

This paper is organized as follows: In Section 3, the experimental method used in this research is described. Proposed flow regime classifiers are discussed in Section 4. In Section 5, the results and discussion of the analyzed data are presented, and finally, conclusions and future work are given in Section 6.

2. Data collection

2.1. The continuous-wave Doppler ultrasonic measurement principles

The ultrasonic sensor used in this research is the Continuous Wave Doppler Ultrasound (CWDU) flowmeter, and its schematic diagram is shown in Fig. 1. The ultrasonic beam continually penetrates the flow in the vertical part of the S-shaped pipe, and the ultrasonic beam correspondingly reacts according to the variant fluid situations (Fig. 1 [a]). The receiver then recognizes the ultrasonic signal's change and converts it into a corresponding waveform digital signal (Fig. 1 [b]) (Nnabuife et al., 2020).

The changing frequency of an acoustic wave when there is a movement or shift between the source and the acoustic receiver, and the frequency shift is proportional to the acoustic source velocity, is referred to as the Doppler shift. Calculating the shift in frequency between the acoustic source and the receiver yields the acoustic source velocity. The CWDU technology involves constantly discharging an acoustic frequency beam from the transducer into the flow and reflecting the sound wave by moving the scattering. Another ultrasonic transducer receives the dispersed acoustic beam, and the flow velocity is computed with the frequency change based on the Doppler effect (Nnabuife et al., 2021).

Fig. 2 depicts and describes the instrumentation procedure required to detect Doppler changes in the received ultrasound: Assuming the transmitted signal is

$$x_t(t) = \varepsilon_t \cos\left(w_s t\right) \tag{2.1}$$

Furthermore, the receiving signal from one of the scatterers is

$$x_t(t) = \varepsilon_r \cos\left(\left\{w_s + w_D\right\}t + \theta_1\right) \tag{2.2}$$

Where $w_s = 2\pi f_s$, $w_R = 2\pi f_R$ and the phase based on the scatterer distance from the phase shifts initiated within the receiver and the transducer is θ_1 (Cobbold, 1989).

Electronically multiplying the two signals yields

$$x_t(t)x_r(t) = \varepsilon_t \varepsilon_r \cos\left(w_s t\right) \cos\left(\left[w_s + w_D\right]t + \theta_1\right) \tag{2.3}$$

$$x_t(t)x_r(t) = \frac{\varepsilon_t \varepsilon_r}{2}\{\cos(w_D t + \theta_1) + \cos(\left[2w_s + w_D\right]t + \theta_1)\} \tag{2.4}$$

As a consequence, the resultant signal is ultimately low-pass filtered to remove the $2f_s$ source frequency while retaining just the Doppler signal (Cobbold, 1989).

$$x_D(t) = \frac{\varepsilon_t \varepsilon_r}{2} \cos(w_D t + \theta_1) \tag{2.5}$$

However, because the received ultrasound signal has reflected ultrasound of larger amplitude than the signal backscattered from the moving scatterer, further signal processing may be required. Doppler shifts at a low frequency in this form of reflected ultrasound. As a result, to eliminate this anomaly, band-pass filtering may be required. (Cobbold, 1989).

2.2. Test rig and experimental procedure

The research was conducted at Cranfield University's Process Systems Engineering Laboratory equipped with state of art industrial-scale multiphase and single-phase flow systems to study the complex behavior of fluids in process plants and oil and gas production. The ultrasonic sensor and the auxiliary instruments are shown in displayed as Fig. 3 (Nnabuife et al., 2020).

The fully automated unique three pressure facility contained 50 m long 4" and 2" flowline and 11m high riser systems and could deliver 140 m^3/hr and 1400 m^3/hr of water and oil. The flow facility could also supply up to 40m/s of air in a 4" pipeline system. The test facility was rated up to 20 barg but currently, the capacity was limited to 7 barg maximum air pressure from the compressors. The flow facility was managed and controlled by DeltaV a Fieldbus based supervisory, control, and data acquisition (SCADA) software produced by Emerson process management.

The 2-inch S-shaped test facility used in this experiment had a 54.8-mm internal diameter, 40-m length, and 1.5-m down comer as shown in Fig. 4. The 2-inch S-shaped pipeline-riser test section has a transparent tube for flow regime observation (Nnabuife et al., 2020).

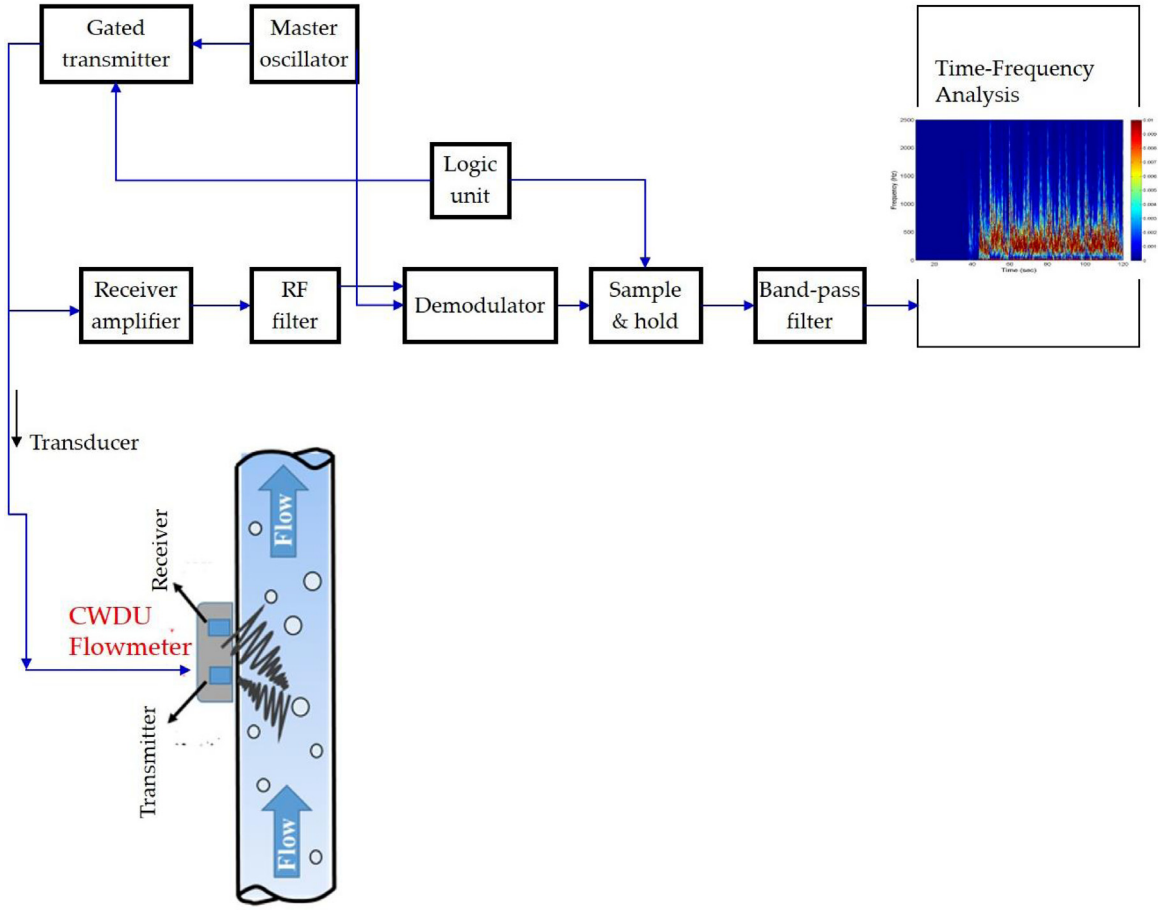


Fig. 2. Hardware block diagram of a continuous-wave Doppler ultrasound system (Nnabuife, Sharma, et al., 2021)

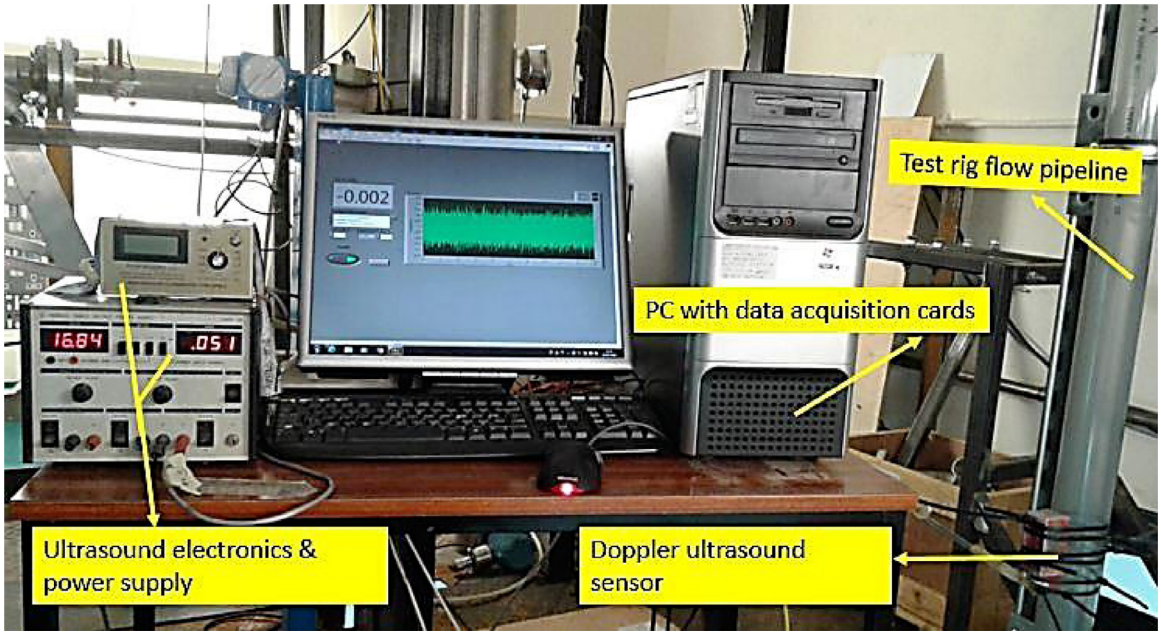


Fig. 3. The ultrasonic sensor and the auxiliary instruments used in this research (Nnabuife et al., 2020).

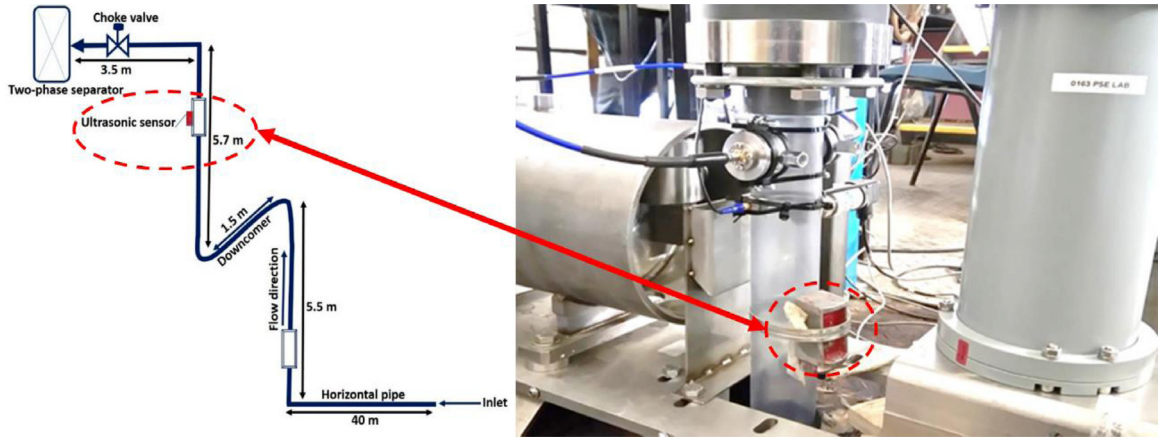


Fig. 4. Schematic diagram of the 2-inch S-shaped flow loop test (Nnabuife et al., 2020; Nnabuife, Pilario, et al., 2019).

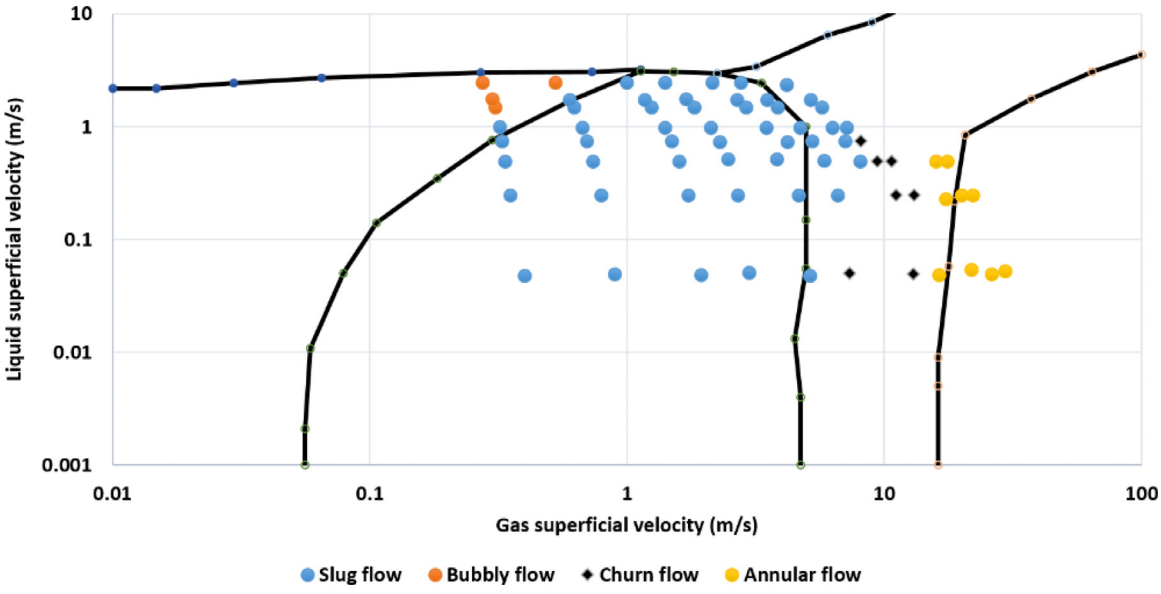


Fig. 5. Flow regime map for 2-inch S-shaped riser.

A non-intrusive clamped-on CWDU with ±10V excitation voltage, operating at a 500 kHz carrier frequency, was attached S-shaped riser top-side section as shown in Fig. 4.The ultrasound beam incident angle was 58° to flow direction on the S-shaped riser. To make the ultrasound energy transmission easier, a gel coupling agent was used between the Doppler transducer and the pipe wall. The continuous-wave Doppler ultrasonic flow meter was employed to record the Doppler voltage signals for further analysis (Nnabuife et al., 2019). Fig. 5 depicts the flow regime map determined from the test facility for the various flow conditions examined in this research.

The process variable measured by the Doppler ultrasonic sensor was flow average velocity. Based on the flow velocity range and the pipe scale, it was evaluated that the flow velocity varies at a frequency no more than 2 kHz. Therefore, in the LabVIEW data acquisition technology, a 10 kHz sampling frequency was appropriate to the Nyquist criterion. This was five times the measured upper limit of the flow velocity frequency variation (Nnabuife et al., 2019).

2.3. Ground-truth dataset

The data used in this research is the flow regime Doppler ultrasonic signal dataset in an S-shaped riser from the open-access Cranfield Online Research Data (CORD) record (Nnabuife et al., 2019). Its reliability has been verified through several studies (Kuang et al., 2021; Nnabuife et al., 2020; Nnabuife et al., 2019). The dataset (Nnabuife et al., 2019) consists of 125 equally-sized subsets of data each corresponding to one experimental record. Each experimental record contains 1.3 million recorded data points.

The four flow regimes have been labeled with integers “1”, “2”, “3”, and “4” corresponding to slugging flow, bubbly flow, churn flow, and annular flow, respectively. Notably, that the integer labels are further encoded with the one-hot format in the training process to avoid gradient confusion.

3. Data preprocessing

This research adopts a single window segmentation algorithm (see Fig. 6 and Algorithm 1) to augment the overall dataset. The yellow window segments a part of the ultrasonic signal as the input for further flow regime classifiers. Nnabuife et al. (2020) found about 50k data points ($length_{window}$) came a 96% identification accuracy, which indicates even a part of the individual record can provide sufficient information for flow regime identification (Nnabuife et al., 2020). However, the state-of-the-art researches require at least 50k data points as one minimum input length of ultrasonic signal. Nnabuife et al. (2020) found the 50k data points of $length_{window}$ is the limitation of their DNN-based

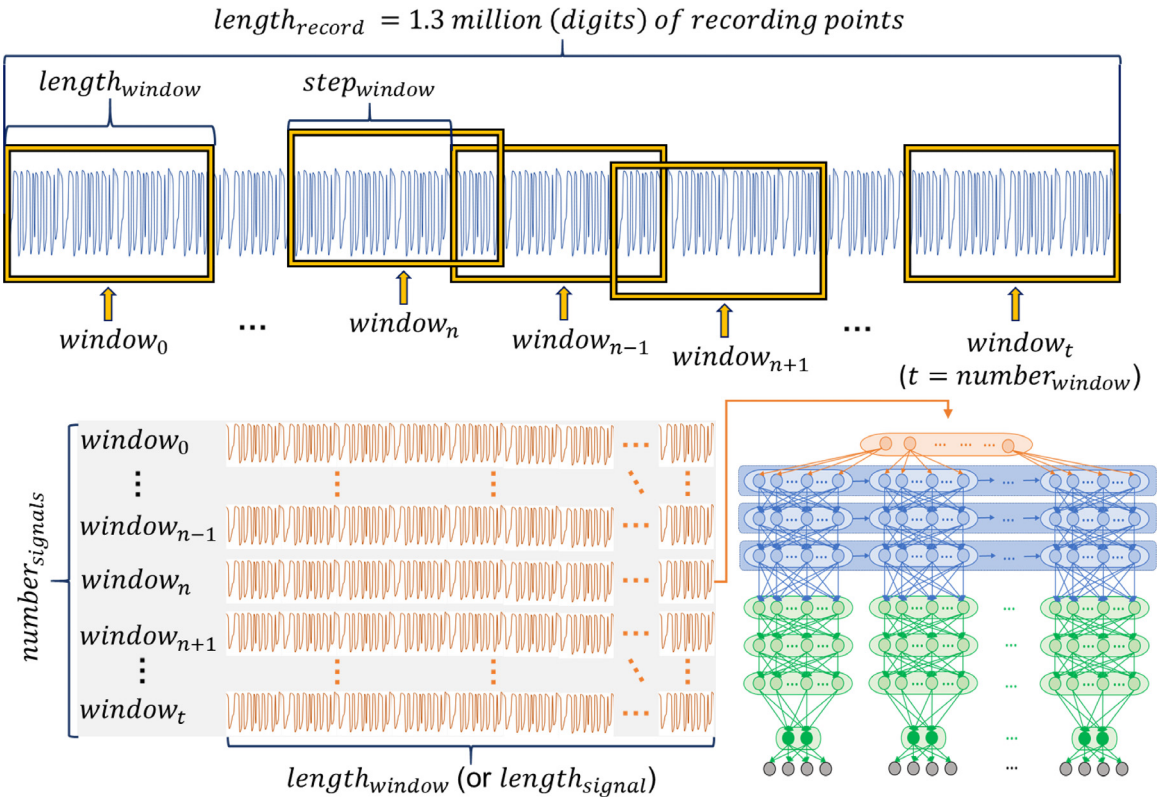


Fig. 6. The process of the single-window segmentation-based data augmentation. The blue (top) and orange (bottom-left) waveform signals respectively refer to the flow ultrasonic signal and the augmented data. The bottom-right neural network represents all the flow regime classifiers used in this research. $\boldsymbol{length_{input}}$ corresponds to the length of the classifiers' input, which equals to $\boldsymbol{length_{window}}$.

Table 1
The setting details of the three data augmentation schemes. The length of the record ($\boldsymbol{length_{record}}$) is 1.3 million (digits) of data points, and the sample frequency is 10kHz. The three augmentation schemes correspond to sensing durations of 1.6384, 0.8192, and 0.4096 seconds respectively.

Augmentation scheme	$length_{window}$	$step_{window}$	$number_{signals}$
Unit	digits	digits	samples
A	16,384	4,096	39,125
B	8,192	4,096	39,375
C	4,096	4,096	39,500

classifier (Nnabuife et al., 2020). The minimum $length_{window}$ in (Kuang et al., 2020) is also 50k. The minimum input length for classifier directly correspond to the classifier's on-time property. The length of the signal input to the neural network represents the amount of data required by the classifier, and the direct manifestation of such amount of data on the ultrasonic signal is the time length of the signal. Scheme A, B, and C separately use 16,384, 8,192, and 4,096 digits as input lengths (or $length_{record}$), which correspond to the signal lengths of 1.6394, 0.8192, and 0.4096 seconds regarding the 10kHz frequency (see the footnotes of Table 1). The reasons for choosing the three signal lengths ($length_{window}$) as follows:

(1) The signal lengths in Scheme A, B, and C are shorter than the state-of-the-art. The length settings recommended in Kuang et al. (2021), Nnabuife et al. (2020), and Nnabuife, Kuang et al. (2021) are 325,000, 10,000, and 30,000, and the corresponding time durations are 3.25, 1, and 3 seconds, respectively. Scheme A has a shorter time duration than Nnabuife et al. (2020) and Nnabuife, Kuang et al. (2021), but longer than Kuang et al. (2021). As the first scheme and the longest time duration, scheme A verifies the effectiveness of the preprocessing (Section 3). Specifically, the corresponding classifier cannot prove its advance if the classifier does not show a competitive result to the state-of-the-art in Scheme A. Both Schemes B and C have shorter time durations than state-of-the-art. These two schemes aim to test the novelty of real-time. Scheme C is committed to decreasing the signal duration further to explore the performance of each classifier on a very short signal duration.

(2) The $length_{window}$ in Scheme A, B, and C are multiples of 2, which are 2 to the power of 14, 13, and 12, respectively. The purpose is to use Fast Fourier Transform (FFT) for faster and more efficient preprocessing. Fourier transform is a complicated process, while the FFT can save a lot of computing power and time. In Scheme C, the $length_{window}$ and the $step_{window}$ are equal, which is equivalent to no overlaps between all samples. Further, reducing the window length can cause significant information loss in the preprocessing, which is pointless for data preprocessing design. Furthermore, the experiment in Section 5 proves that the signal duration of Scheme C is too short for the flow regimes identification.

Combining the above two points, Scheme A, B and C use long, short, and very short time durations to construct the signal lengths experimented by each classifier.

Eq. (1) depicts the qualitative relationship between the robustness and the length of the signal ($length_{signal}$) and the number of signals ($number_{signals}$). Notably, the number of signals ($number_{signals}$) refers to the available number of training signals. This research limits the $number_{signals}$ to a value around 40k data points using the length of window shifting ($step_{window}$), which is an enough large value to achieve a convinced identification result. Table 1 shows the progressive settings data augmentation schemes, and Eq. (2) depicts the numerical

Table 2
the specifications for all the flow regime classifiers in the proposed CRNN benchmark. Experiment index provides an index for all fourteen experiments, which also corresponds to the "idx" in Table 3. Notably, Experiment No.14 is a special case which has been discussed in Section 5.2. The "Classifier type" indicates the applied classifier. The "Augmentation scheme" corresponds to the Scheme A, B, and C in Table 1. The "Usage" refers to the purpose of different classifiers, where "C" stands for the comparison-classifiers, and "T" stands for testing-classifiers. The "Axis" refers to the main learning strategy, where "T" stands for the recurrent structures and "S" stands for the convolutional structures. The "Additional explanation" includes some key information for different models.

Experimental scheme	Experimental index (No.)	Classifier type	Augmentation scheme	Key characteristics		
				Usage	Axis	Additional explanations
Scheme "reference"	1	FCN	A	C	S	Only use convolutional structures
	2		B			
	3		C			
	4	RNN	A	C	T	Only use the recurrent layers
Scheme 1	5	CRNN1.x	A	T	S+T	A simple dimension alignment among tensors
	6		B			
Scheme 2	7	CRNN2.1	A	T	S+T	PTS vector-set + complicated CRNN
	8		B			
	9	CRNN2.2	A	T	S+T	PTS vector-set + simple CRNN
	10		B			
Scheme 3	11	CRNN3.x	A	T	S+T	Multi-scale CRNN
	12		B			
	13		C			

relationship among $length_{signal}$, the length of window ($length_{window}$), $step_{window}$, and $number_{signals}$.

Algorithm 1: The single window-based data augmentation

Input: Ground expermental signals [35]: $Records$
The number of experimental records: $number_{records} = 125$
The length of input records: $length_{rwcords} = 1,300,000\ digits$
The length of window: $length_{window} \leftarrow$ Tab.2
The length of window shifting: $step_{window} \leftarrow$ Tab.2
The number of augmented signals: $number_{signals}$

Output: Augmentated flow regime dataset: $Simples$.

```
for i_r in range(number_records) do
    record_curr = Records[i_r, :]
    number_signals ← Eq.(2) ← (length_window, step_window)

    for i_w in range(number_signals) do
        signal_curr = record_curr[i_w * step_window : (i_w + 1) * step_window]
        Simples ← append (signal_curr)
    end

end
```

$$\begin{cases} robustness \propto length_{signal} \\ robustness \propto 1/number_{signals} \end{cases} \tag{1}$$

$$number_{signals} = \frac{length_{record} - length_{window}}{step_{window}} \tag{2}$$

3. Experimental designs

Convolutional Neural Networks (CNN) are effective in feature extraction, which provides a means for overcoming the low signal-noise ratio challenge. Nnabuife et al. (2019), Nnabuife et al. (2020), and Kuang et al. (2020) utilize PCA and Fourier transformation to handle the challenge of high noise-signal-ratio, which complicate the overall procedure also require prior optimizations (not E2E) (Kuang et al., 2020; Nnabuife et al., 2020; Nnabuife et al., 2019). Furthermore, the parameter-sharing (of CNN) can suppress the overfitting risk to achieve a better identification performance. However, the depth increases in convolutional structure inevitably increase the model complexity and gradient vanishing risk. On the other aspect, Recurrent Neural Networks (RNN) provide an approach to implicitly connect information along the time-axis (Zaremba et al., 2014). The recurrent layer can map the input shape to a desired output shape (Ghavamian and Simone, 2019). However, these implicit connections (within a recurrent structure) inverse-correlate to the timestamp extension (Zaremba et al., 2014).

This research uses convolutional structures to create a pseudo-time-series (PTS), then utilizes recurrent layers to dig the implicit connections along the PTS. The PTS can be understood as an artificial time-axis, which is the output of convolutional neural networks (CNNs) feature extraction along the real time-axis (the named pseudo-time-series). The interesting part is the inevitable consequence for receptive field increase in CNNs can be simply handled using RNN with the CNNs-generated PTS axis. Therefore, the key mission for this research focuses on finding the CRNN classifier which can achieve high identification accuracy with a very simple model-complexity.

This research involves three CRNN schemes described in Table 2. Notably, a gradual strategy is adopted for applying different augmentation schemes to the classifiers. The gradual strategy refers to "only adopts a shorter $length_{signal}$ (augmentation scheme) if the longer $length_{signal}$ (augmentation scheme) achieves proper performance". Section 4.1 discusses the fully convolutional network (FCN)-based classifier (without any recurrent structure). Section 4.2 depicts the only RNN-based classifier. They represent two extreme design with only CNN and RNN as reference cases. Section 4.3 then considers the CRNN classifiers (three CRNN schemes involved).

3.1. FCN-based flow regime classifier

This section attempts to identify the flow regime with only convolutional structures (see Fig. 7). The FCN-based flow regime classifier can be divided into two parts, the feature extraction part and the classifier part.

The feature extraction part can extract the input 1D time-series data into a compact feature. The expansion block (ConvBlock-e) parallelly expands the 1D input into multiple 1D vectors (Fig. 8 depicts ConvBlock-e's structure). The "e" stands for the "expand" in ConvBlock-e. The subsequent feature extraction blocks (ConvBlock-fX) gradually extract valid features by increasing the depth along six convolution blocks. The structure of ConvBlock-fX is depicted in Fig. 7, where "f" stands for "feature extraction" in ConvBlock-fX. ConvBlock-fX adopts a structure similar to the residual neural network (ResNet) (He et al., 2015). The ResNet design can achieve better convergence and higher accuracy (Yao et al., 2020). The "Add layer" creates a "highway" in ConvBlock-fX, which boosts the gradient's backpropagation to avoid the vanishing gradients.

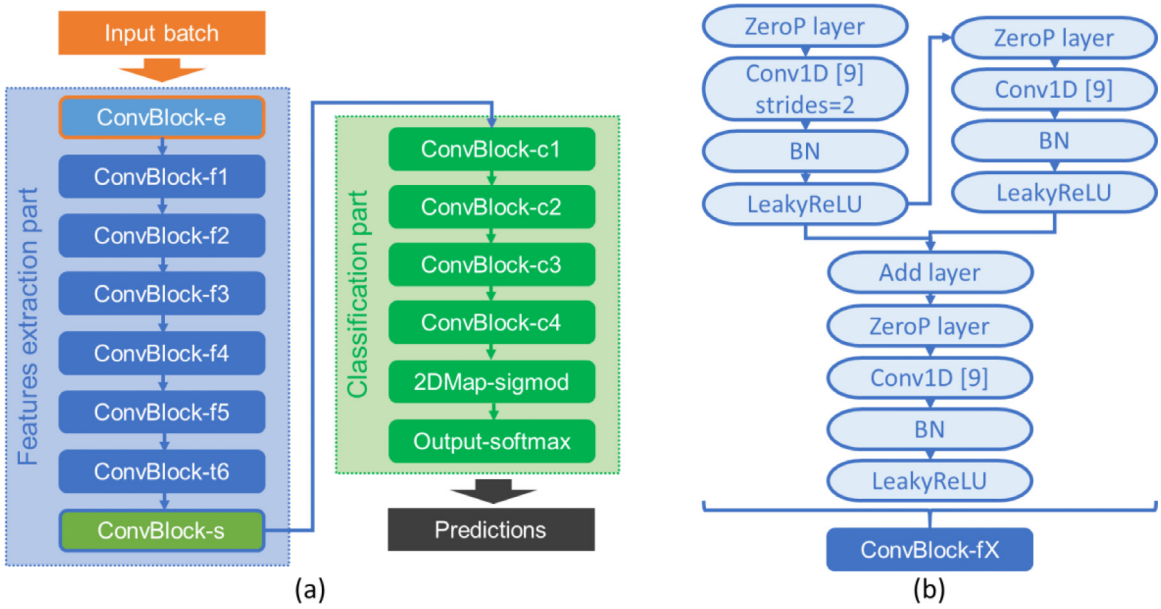


Fig. 7. The detailed structure of the proposed FCN-based flow regime classifier. (a) illustrates the overall layout of the FCN-based classifier. (b) depicts the structure of the ConvBlock-fX. The structures of the ConvBlock-e, ConvBlock-s, and ConvBlock-cX are placed in Fig. 8. For all Fig.s in this paper, the 'X' marks, rectangular boxes, rounded rectangular boxes, elliptical boxes, solid lines represent the block index, in/output data, neural network blocks, layers, and inside data flow, respectively. ZeroP, Conv1D, BN, LeakyReLU refer to the zero-padding layer, 1D convolutional layer, batch normalize layer, and leaky rectified linear unit (LeakyReLU) activation function, respectively. The X inside "[X]" after the Conv1D (or Conv2D) refers to the kernel size, and the "strides=X" refers to the stride's length equals to X for kernel shifting.

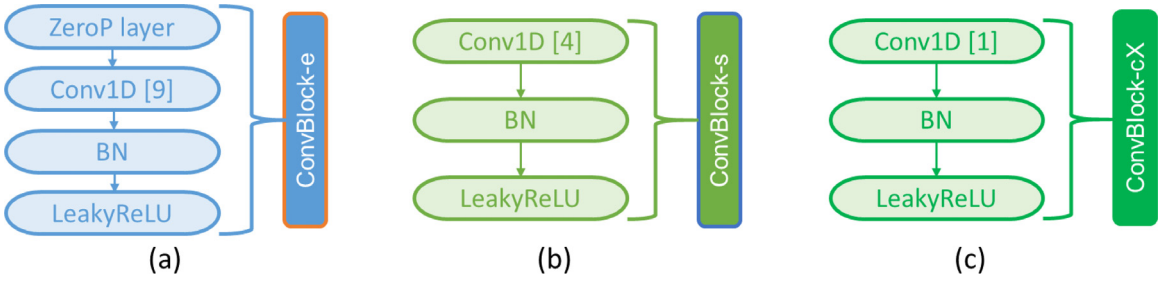


Fig. 8. The structure of the ConvBlock-e, ConvBlock-s, and ConvBlock-cX in this research.

ConvBlock-fX uses the convolutional layers with strides equals to two instead of the pooling layer to suppress information loss in the feature extraction part. Finally, the stretch block (ConvBlock-s) (Fig. 8 [b]) stretches the extracted features back into 1D form, where "s" stands for the "stretch." Notably, ConvBlock-s' output is a compact feature with a shorter length than the input data.

Fig. 7 shows that the FCN structure uses five times ConvBlock-fX block, and each ConvBlock-fX performs a convolution operation with a stride of two. Furthermore, the dimensional changes introduced by stride are also applicable to the RCNN design in Section 4.3. The receptive field of each node in ConvBlock-t6 in ConvBlock-f1 is about 1,700 digits. Assuming that the convolutional kernel of a stride-convolution is "m" and stride is "s", the size of the receptive field ("r") in this convolutional layer is "$r = (m - s) * s + m$". Further increasing the receptive field can increase the neural network depth, which increases the risk of the gradient vanishing. It is noteworthy that, in order to suppress the gradient vanishing, this study applies a connected structure like the ResNet (see Fig. 7(b)). Therefore, it is unnecessary to increase the receptive field further because of the risk of gradient vanishing. On the other hand, reducing the receptive field to reduce the risk of gradient vanishing is also inappropriate. In the RCNN structure in Section 4.3, the convolution structure also conducts the function of dimensional change. Stride-convolution increases the receptive field while also compressing the dimension of the tensor. When stride is equal to 2, one stride-convolution can compress the tensor to its 1/2. The FCN classifier and RCNN classifier use five times stride-convolutions. Therefore, the data is compressed by 2^5 (or 32) times. If the number of stride-convolution is reduced, the tensor introduced to the classification part becomes very large. Although this does not cause too much problem for the FCN network, the RNN layer in the CRNN will significantly reduce the computational efficiency.

The classification block (ConvBlock-cX) in the classification part (Fig. 7) is a convolution block that uses the convolution kernel equals to one, operating like the fully connected layer. The final output-softmax layer uses a vector length of four to output the prediction. Output-softmax is composed of a flat layer followed by a fully connected layer with four neurons, and the activation function is the softmax logical regression.

The FCN-based flow regime classifier implements a two-dimensional (2D) mapping from ConvBlock-e to 2DMap-sigmoid (Fig. 7 [b]). Using the sigmoid activation, the four flow regimes are mapped into a 2D square space with each edge equal to one, which is discussed in detail in Section 5.3. Notably, all proposed classifiers adopt the 2DMap-sigmoid and Output-softmax structure as the output port. Therefore, the classifiers firstly map the flow regime data to a 2D square space and then conduct the classification task.

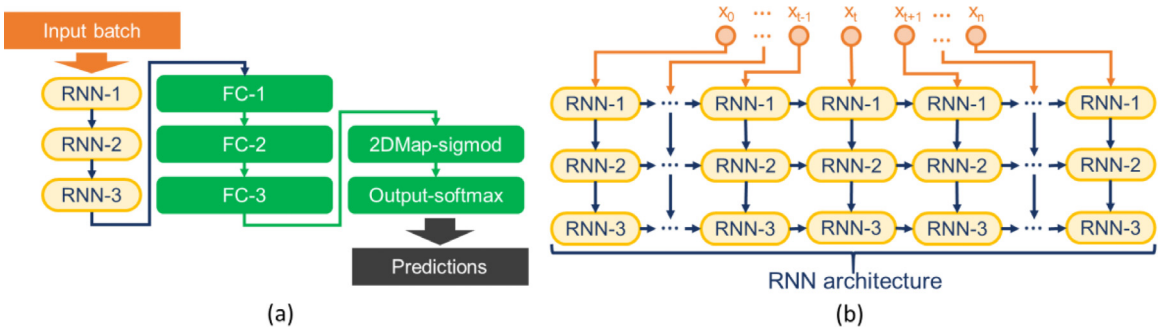


Fig. 9. The detailed structure of the proposed RNN-based flow regime classifier. (a) depicts the overall layout of the RNN-based classifier. The RNN-X refers to the recurrent neural network block with index X. FC-X refers to the fully connected block, composites with fully connected layer, and LeakyReLU activation function.

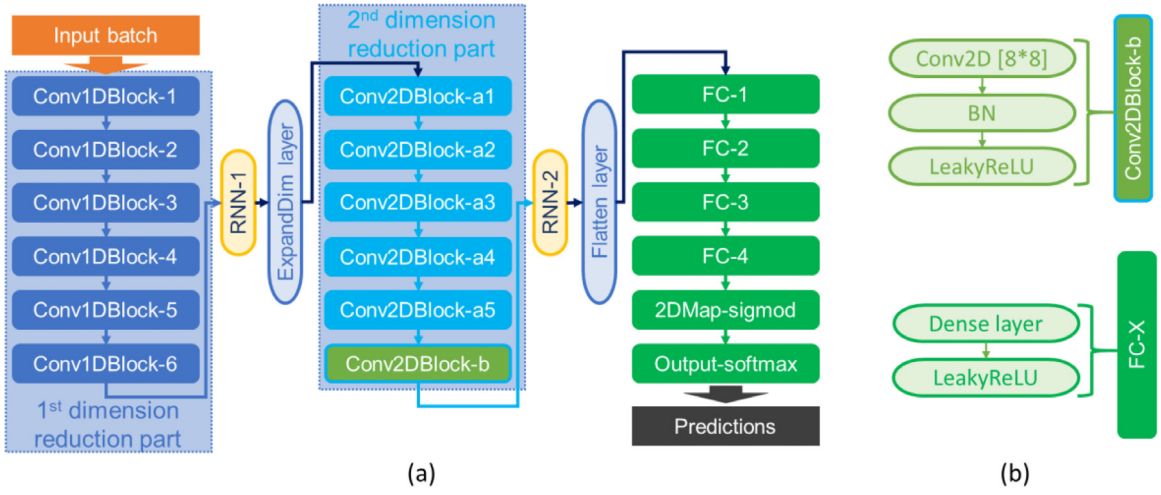


Fig. 10. The detailed structure of the proposed CRNN1.x-based flow regime classifier. (a) depicts the overall layout of the CRNN1.x classifier. (b) refers to the structure of Conv2DBlock-b, FC-X in this research. The structure of Conv1DBlock-X and Conv2DBlock-aX have been illustrated in Fig. 11.

3.2. RNN-based flow regime classifier

This section focuses on the time-series method. RNN is a popular representation of time-series methods. The RNN-based flow regime classifier (Fig. 9 [a]) only utilizes the recurrent structures. The input 1D Doppler signal passes through three RNN-blocks, then enters the three fully connected layers. Fig. 9 (b) highlights the operating process of the RNN blocks in Fig. 9 (a). The orange circles (Fig. 9 [b] top) represent a single sample in the input batch (Fig. 9 [a]). x_0, x_{t-1}, x_t, x_{t+1}, and x_n are the data point at the start (0), $t-1$, t, $t+1$, and end timestamp (n), respectively. There are two types of RNN blocks used in this research according to the sequence return formation. The first type returns the entire timestamp results, while the second type only returns the end timestamp result. RNN-1 and RNN-2 blocks belong to the first type, and RNN-3 block belongs to the second type. The columns (Fig. 9 [b]) represent the forward and backward propagation inside the RNN-based classifier, and the rows (Fig. 9 [b]) are the propagation along the time axis.

3.3. CRNN-based flow regime classifiers

3.3.1. CRNN1.x: dimensional alignment

This section discusses the CRNN1.x-based flow regime classifier, which is a simple combination using convolutional and recurrent structures. Notably, the FCN-based classifier attempts to decrease the model complexity by parameter-sharing strategy and increases the receptive field by increasing depth (also increase the model complexity). Such conflicts decrease the efficiency of online flow regime prediction. On the other side, the RNN-based classifier attempts to utilize the time-correlation, but the RNN-3 block loses a lot of information because of the significant dimension change. Moreover, the time-correlation established in the RNN-based classifier is actually weak because of the long timestamp range. For the timestamp n output, only the timestamp n-1 is directly related. As a result, striking a balance between convolutional and recurrent structures is difficult.

The input and output data tensors of convolutional and recurrent layers both have three dimensions, which is indicated with batch size, timestamp length, and dimension, respectively. The convolutional layers of the CRNN1.0 classifier mainly work for the dimension reduction (Fig. 10 [a]), while the recurrent structure focus on time-correlation. Fig. 11 (a) depicts the structure of Conv1DBlock-X, which also adopts a structure similar to the ResNet (He et al., 2015). Each Conv1DBlock-X is composed of a convolutional layer of strides larger than one (Fig. 11 [a] left), and a convolutional layer of stride equals to one on the right of (Fig. 11 [a] right). RNN-1 learns the time-correlation from the Conv1DBlock-6. The 2nd dimension reduction part utilizes the 2D convolutional blocks (Conv2DBlock-aX) (Fig. 8 [a]) imitates the image processing, uses 2D convolutional layers to compress the results of RNN-1 further. Conv2DBlock-b (Fig. 10[b] top) stretches the feature back into a 1D form after the five Conv2DBlock-aX. This one-dimensional feature is input into the RNN-2 structure, and then FC-X (Fig. 10 [b] bottom) performs dimension reduction and finally outputs the prediction vector.

3.3.2. CRNN2.x: pseudo-time-series vector-set

The section discusses the CRNN2.x flow regime classifier. The CRNN1.x classifier equips low computational efficiency, convolutional layers play the role of dimension transformer for inside data flow. However, the essence of a convolutional operator is a depth expansion process using parallel convolution kernels. In other words, convolution is a

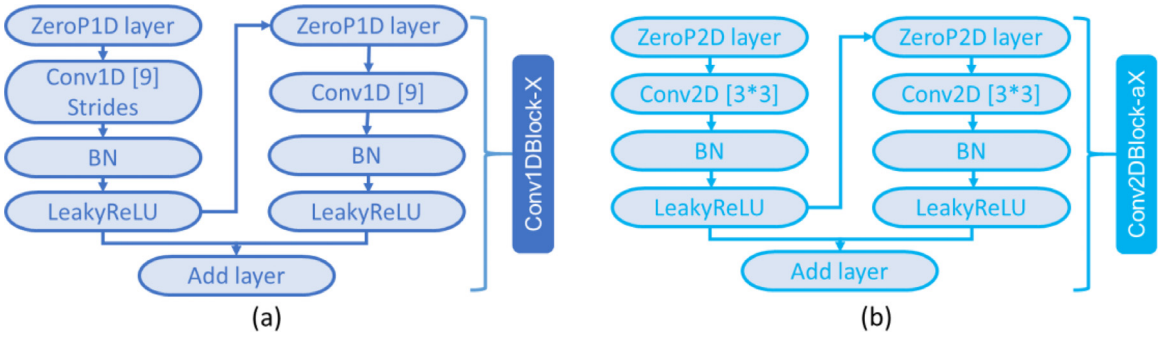


Fig. 11. The structures of Conv1DBlock-X and Conv2DBlock-aX in this research

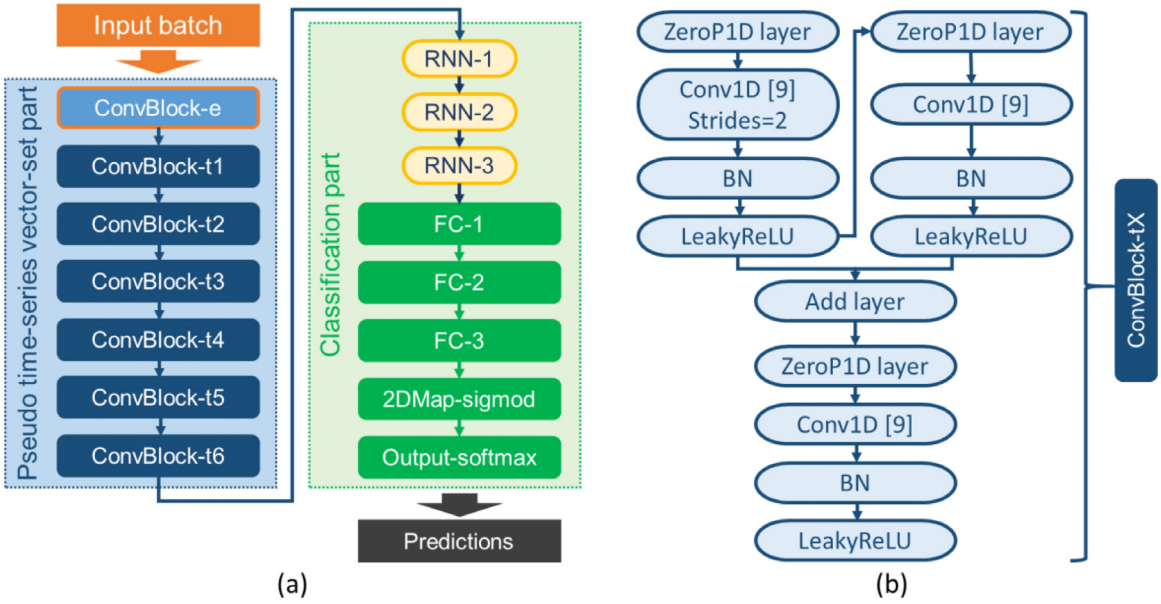


Fig. 12. The detailed structure of the proposed CRNN2.x-based flow regime classifier. (a) depicts the overall layout of the CRNN2.x classifier. (b) refers to the structure of ConvBlock-tX

mapping operator that maps data from input to output space. CRNN2.x proposes a more efficient convolutional and recurrent structures.

CRNN2.x can be divided into two parts: PTS vector-set part (Fig. 12 [a]) and classification (Fig. 12 [a]). The PTS vector-set part imitates the structure of FCN (Section 4.1). ConvBlock-tX (Fig. 12 [b]) is a 1D convolutional layers block, where t stands for transformation. ConvBlock-tX also uses a structure similar to the ResNet (He et al., 2015). Notably, the stride of the first layer of ConvBlock-tX equals two to replace the pooling operator. The depth gradually increases from ConvBlock-t1 to ConvBlock-t6. Compared with the input data, the output of ConvBlock-t6 is a vector-set whose size is greatly shorted. This research supposes the size-axis sets as the timestamp axis and the depth-axis sets as the vector-axis. The PTS vector-set part's output is a vector-set compressed along the timestamp-axis but extended along the depth-axis. The PTS vector-set consists of multiple PTS vectors distributed along the pseudo-timestamps-axis. The classification part inputs the PTS vector-set into the classifier composed of three RNN blocks, three FC-X blocks, 2DMap-sigmoid, and Output-softmax.

CRNN3.x: multi-scale model

This section discusses the CRNN3.0 multi-scales CRNN-based flow regime classifier. Fig. 13 shows the structure of CRNN3.x, which can be divided into multi-scale virtual time-series vector-set part and the classification part.

The multi-scale PTS vector-set part adopts a multi-scale convolutional structure. A deep convolutional layers layout can cause significant information loss, but a shallow layout cannot provide enough computational complexity. They respectively use 20, 25, and 19 convolutional layers to conduct the dimension transformation in the above FCN, CRNN1.x, CRNN2.x-based classifiers, while CRNN3.x only requires nine convolutional layers. CRNN3.x parallelly achieves three same-sized PST-vector-set. Thus, the depth of each convolutional layer scale in each channel is only three blocks. The structure of Conv1DBlock-X is shown in Fig. 14 (a) and (b). The classification part comprises four CRNN blocks (Fig. 14 [d]), which contains one RNN block and two Conv1DBlock-X blocks. After CRNNBlock-X, RNN-5 and a fully connected layer are added to complete the classification.

4. Results and discussions

4.1. The proposed CRNN-based flow regime identification benchmark

The hardware utilized in this research was the Intel E5-2620 v4 (Broadwell) Central Processing Units (CPUs) of 16 CPU cores, the memory of 64GB, and the Tesla K80 Graphics Processing Unit (GPU) card. This research developed on the Linux 18.04 system and the involved software is Python3.7, Tensorflow 2.0, Numpy, and Matplotlib.

This study keeps the hyperparameters constant on different experiments to eliminate the influence of hyperparameter adjustments. The purpose of the experiments is to compare the performance of the different RNN-based classifiers proposed in Section 4 on flow regime

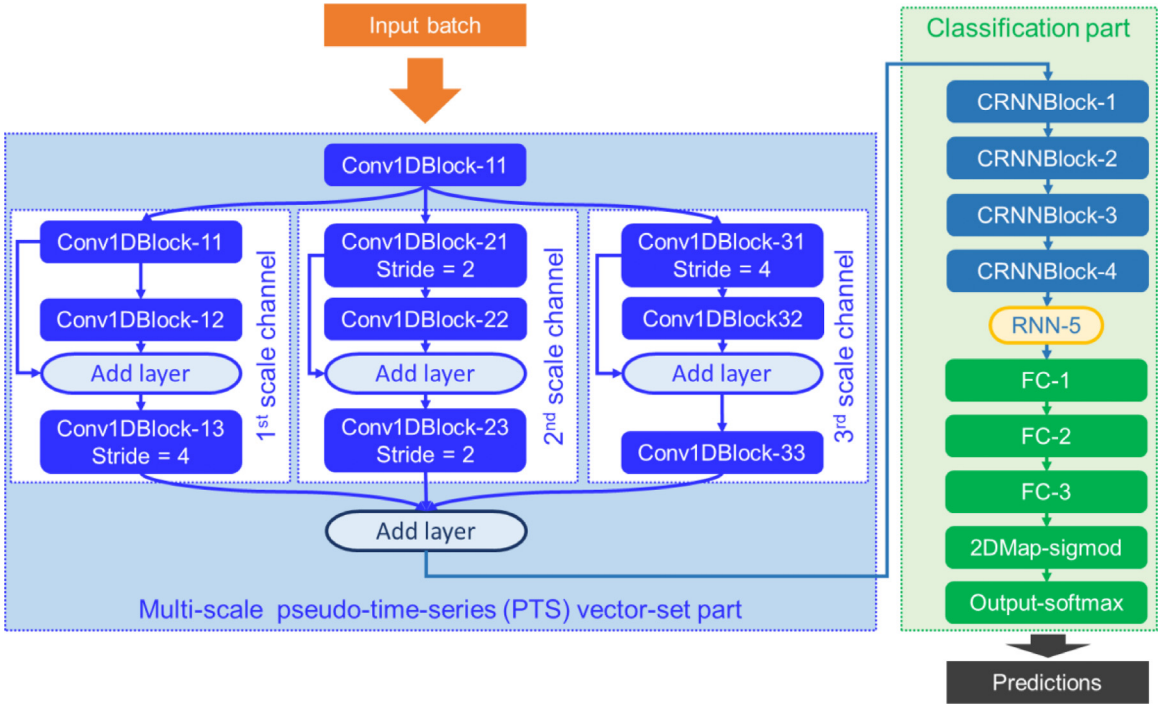


Fig. 13. The detailed structure of the proposed CRNN3.x-based flow regime classifier

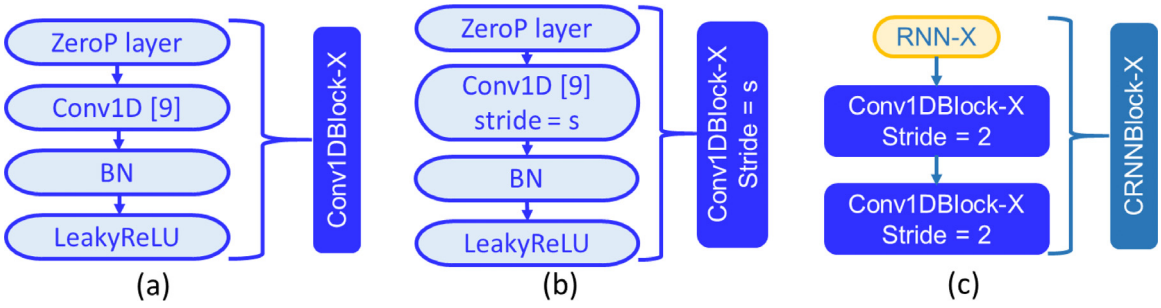


Fig. 14. The structure of Conv1DBlock-X, Conv1DBlock-X (with stride equals to s), and CRNNBlock-X in this research

identification. The setting of hyperparameters will have a great impact on the experimental results. The settings and discussions of the hyperparameters are as follows:

(1) The $number_{input}$ represents the size of the sample space. Deep learning is usually very sensitive to the size of the sample space. The experiments limit the sample space obtained by the different schemes in Section 3 to a small range, 39K to 39.5K samples. The recommended sample space size from Nnabuife et al. (2021) is 12k, so this study assumes that a sample space size of about 39k is a reasonable setting, and the results in Table 3 also support this hypothesis.

(2) This study uses the one-hot encoding method for annotating different flow regimes. One-hot decomposes the multi-classification task into multiple binary classification tasks. For example, the four flow regimes (slugging flow, bubbly flow, churn flow, and annular flow) in Section 2 are marked using integers "1", "2", "3", and "4". However, they can bring confusion to the gradient. For example, a predicted value of 1.5 has the same gradient to annotation "1" and "2". The one-hot encoding transforms the annotations along the following patterns, "1->(1,0,0,0)", "2->(0,1,0,0)", "3->(0,0,1,0)", and "4->(0,0,0,1)". Thus, any flow regime is recognized as a binary classification task (between 0 and 1). The one-hot encoding can accelerate the model convergence and avoid gradient confusion.

(3) All the data from Section 3 has been randomly shuffled before performing any experiment.

(4) The shuffled samples are divided into the training set, testing set, and validation set according to the ratio of 60%, 20%, and 20%. Specifically, 0 to 60% is used as the training set, 60% to 80% is used as the testing set, and 80% to 100% is used as the validation set. It is noteworthy that the verification of the proposed flow regime classifier is conducted using the testing and verification sets. Specifically, the goal of the training set is to provide experience for training the classifiers. The entire training process repeatedly iterates on the training set, and once learning is also called an epoch. The epoch does return not only the training results but also the evaluation results using the testing set. The testing set contains the data independent from the training set, which can be regarded as new data different from the training set. The results of the testing set are used to evaluate the generalization ability and fitting condition of the classifier, while the fitting condition mainly refers to the level of overfitting and underfitting. However, the testing set is also involved in the general training process because the testing set has been used to evaluate the general training performance of the classifier. Therefore, this study

Table 3
The benchmark of the CRNN-based flow regime classifiers. idx, acc_{train}, acc_{test}, acc_{valid}, $loss_{train}$, $loss_{test}$, mse_{train}, mse_{test}, and complexity respectively refer to the experiment index (corresponding to Table 2), training accuracy, testing accuracy, validation accuracy, training loss, testing loss, training MSE), testing MSE, and the computational graph complexity. "%" stands for the percentage. "/" stands for the non-applicable. The bold values refer to the best record in this research. The shadow refers to the highlighted result, which achieves the best performance balanced between model's complexity and accuracy. Nnabuife et al. (2019); Nnabuife et al. (2020), and Kuang et al. (2020) refer to the state-of-the-art. Experiment No.14 further trains the model from Experiment No.11, so the epoch callback goes to 300 instead of 200. Experiment No.14 achieves the best result in this research, which has been discussed in Section 5.2.

idx unit	acc_{train} %	acc_{test} %	acc_{valid} %	$loss_{train}$ /	$loss_{test}$ /	mse_{train} /	mse_{test} /	complexity parameters
1	99.98	97.13	97.30	0.2112	0.2870	0.0160	0.0256	7,436,654
2	99.85	86.38	84.70	0.2522	0.5181	0.0189	0.0620	7,436,654
3	99.75	72.53	73.42	**0.1889**	1.0150	**0.0138**	**0.0119**	7,436,654
4	36.73	37.14	25.00	1.2873	1.2831	0.1755	0.1750	756,830
5	**100.00**	97.45	97.63	0.2109	**0.2591**	0.0151	0.0234	6,748,974
6	99.92	85.62	86.02	0.2498	0.5612	0.0202	0.0673	4,651,822
7	99.78	**97.59**	**98.06**	0.2052	0.2643	0.0152	0.0230	17,113,606
8	36.20	37.98	25.00	1.2900	1.2773	0.1759	0.1744	14,242,542
9	99.75	97.36	97.25	0.2051	0.2725	0.0146	0.0231	11,933,702
10	99.75	88.39	89.42	0.2433	0.5109	0.0198	0.0588	3,931,918
11	98.47	92.46	93.60	0.2974	0.4424	0.0276	0.0465	117,702
12	66.88	59.75	41.02	0.8058	0.9824	0.1185	0.1391	101,318
13	78.81	60.17	54.51	0.6113	1.0081	0.0818	0.1379	**93,126**
14	*99.30*	*98.26*	*97.80*	*0.0865*	*0.1235*	*0.0048*	*0.0092*	*117,702*
Nnabuife et al. (2019)	85.70	84.60	/	/	/	/	/	/
Nnabuife et al. (2020)	99.01	96.28	96.35	/	/	0.0060	0.0152	/
Kuang et al. (2020)	99.95	99.95	99.54	0.0013	0.0141	3.9175*e-4	0.0019	/

further applies an independent validation set to evaluate the generalization ability of the classifier in the new environment.

(5) The number of training epochs is set to 200 epochs.

(6) The loss function uses the categorical cross-entropy, which is represented by Eq. (3). The $loss_{cce}$, x_{input}, n, y_{gt}, and y_{pred} respectively refer to the categorical cross-entropy, input batch, batch size, ground-truth label and prediction.

(7) The evaluation metrics apply accuracy and mean square error (MSE). Eqs. (4) and (5) depict the accuracy and MSE used in this research, where TP, FP, FN, and TN refer to the number of true positive, false positive, false negative, and true negative predictions from the machine learning aspect. Accuracy is a direct evaluation index, while MSE is a representation of the Euclidean distance between the predictions and the annotations. It is noteworthy that there is no direct connection between MSE and accuracy. For example, if an annotation equals 1, and the prediction equals 0.8, so the accuracy and MSE equal 100% and 0.2. In another case, the prediction becomes 0.6, while the accuracy and MSE equal 100% and 0.4, respectively.

(8) The optimizer uses Adam, and the initial learning rate is set to 0.0001. This study uses the Adam optimizer implemented by TensorFlow Keras, and other parameters are their corresponding default settings.

(9) Batch size is set to 32 samples per batch.

Table 3 lists all the results of the fourteen experiments depicted in Table 2. Table 3 indicates that the highest testing and validation accuracy were obtained in Experiment No.7, while the simplest complexity were achieved in CRNN3.3. FCN, CRNN1.x, CRNN2.x, and CRNN3.x all obtained better results than (Nnabuife et al., 2019) and (Nnabuife et al., 2020) in the augmentation scheme A. However, most of them (except CRNN3.3) are also very complex. A high complexity brings seriously challenge for the actual industrial scenarios, and lead to a high-risk of overfitting. Table 3 depicts that the training accuracy of FCN, CRNN1.x, and CRNN2.x are close to 100%, but the testing and validation accuracy still low for Experiments No. 2, 3, 6, and 10.

Experiment No.11 achieved a better result than all the baselines proposed by Nnabuife et al. (2019), this is significant. Although its testing and validation accuracies are lower than Nnabuife et al., (2020) and Kuang et al. (2020), its complexity is 98.4% lower compared to Experiment No.7, which is considerable. Even Nnabuife et al. (2020) and Kuang et al. (2020) do not consider the complexity, this research calculates that the complexities are approximately between 1,100k and 2,000k according to their classifier's structure (Kuang et al., 2020; Nnabuife et al., 2020). Therefore, the CRNN3.x separately decreased the complexity by approximately 90% and 95%. Furthermore, Nnabuife et al. (2020) consumed about 1,300 epochs to achieve the 97.59% testing accuracy (Nnabuife et al., 2020), and Kuang et al. (2020) spent 500 epochs to achieve the 99.95% testing accuracy (Kuang et al., 2020). The CRNN3.x only conducted 200 epochs. Section 5.1.5 discusses Experiment No.11 in detail. Further evaluation of the CRNN3.x-based flow regime classifier's performance without epoch limitation is proposed in Section 5.2.

$$loss_{cce} = -\frac{1}{n}\sum_{x_{input}}\left[y_{gt}\cdot\ln(y_{pred}) + (1-y_{gt})\cdot\ln(1-y_{pred})\right] \tag{3}$$

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$MSE = \frac{1}{m}\sum_{1}^{m}(y_{pred}-y_{gt})^2 \tag{5}$$

4.1.1. FCN-based flow regime classifiers

Figs. 15,23, and 24 show the FCN-based classifier training's experimental results with the augmentation schemes A, B, and C (Experiment No.1, 2, and 3 in Table 2), respectively. In Experiment No.1, the FCN-based classifier demonstrates a flow regime identification performance lower than Nnabuife et al., (2020) and Nnabuife et al., (2019). In Fig. 15, the FCN-based classifier uses only 32% of the input length of (Nnabuife et al., 2020) and achieves 99.98%, 97.13%, and 97.30% training, testing, and validation accuracy, respectively. Fig. 15 (b) and (c) show very stable learning trends, indicating that the model is stable.

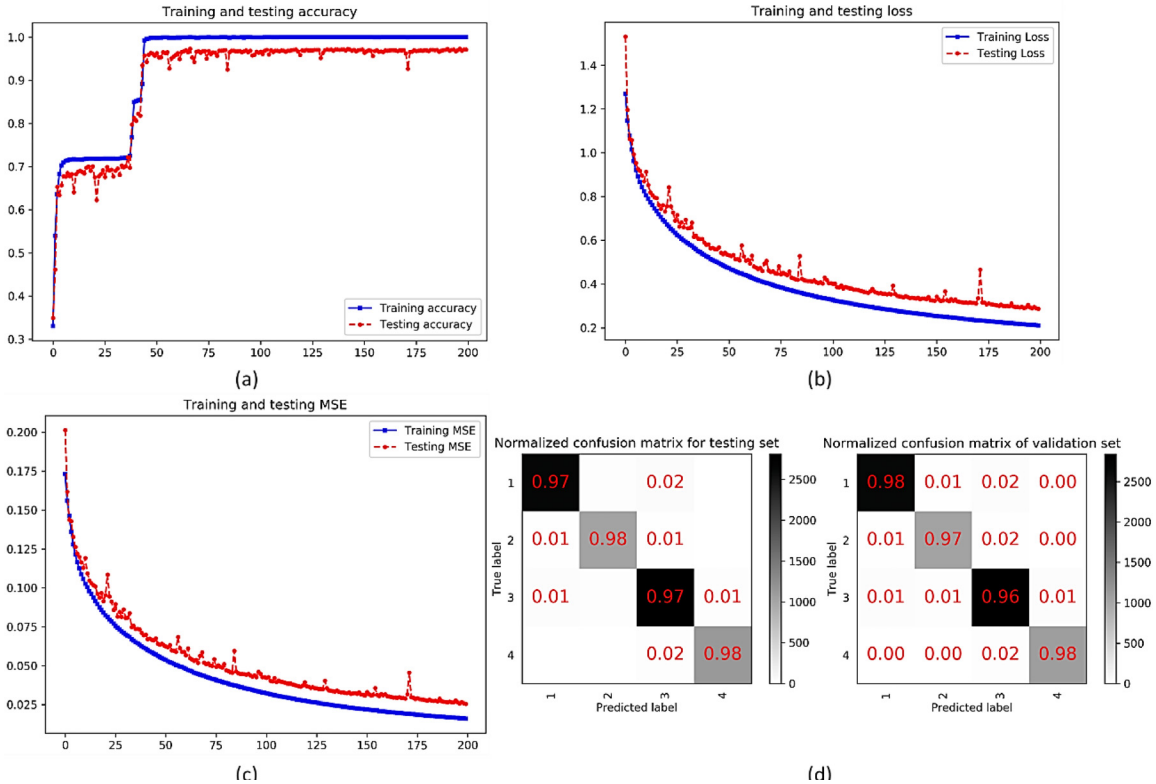


Fig. 15. The detailed experimental records of the FCN-based flow regime classifier using the augmentation scheme A (Experiment No.1 in Table 2 and Table 3). (a), (b), and (c) respectively refer to the training and testing curves of accuracy, loss, and MSE, where the red dash and blue solid curves respectively correspond to the training and testing set. The x-axis in (a), (b), and (c) refers to the epoch. The y-axis refers to accuracy in (a), categorical cross-entropy in (b), and MSE in (c). (d) left and right respectively refer to the testing and validation confusion matrix, where the values correspond to the accuracy, x and y-axis refer to prediction and ground-truth labels. The right color bar indicates the sample's amount to the corresponding color. Notably, all the figures in this paper follows the same meanings as above.

Surprisingly, Fig. 15 (d) depicts that the FCN-based classifier has superior identification accuracy for the four flow regimes, which indicates that the convolutional layers provide significant performance in pattern recognition. However, Fig. 23 and Fig. 24 indicate that the model's performance decreases as the length of the input flow ultrasonic signal decreases, which also verifies Eq. (1) in Section 3. Notably, Figs. 23 and 24 (a), (b), and (c) show that the FCN-based classifier is significantly overfitted. Furthermore, Table 3 mentions that the FCN-based classifier equips 7,436,654 parameters. Although the complex computational graph allows the FCN classifier to achieve high training accuracy when the $length_{input}$ is reduced, the model is easy to overfit.

4.1.2. RNN-based flow regime classifier

Fig. 16 shows the RNN-based classifier results. The accuracy passes 48%, then suddenly drops to and stays at about 35%, which indicates that it is challenging to directly use the RNN. Especially for the long timestamp signals, it brings great computational consumptions (almost impossible). Therefore, the RNN-based classifier uses the second type of RNN block to reduce the amount of calculation before accessing the fully connected layer. The second type of RNN block only returns the result of the last timestamp, which leads to a significant information loss from the early timestamp. The RNN-based classifier only requires 756,830 parameters (Table 3), which decreases complexity by 90% compared to the FCN-based classifier. Therefore, the idea of combining the convolutional structures of FCN with the recurrent structures of RNN becomes very pertinent.

The comparison between Experiment No.4 and Experiment No.1 shows that the convolutional structure has advantages over the recurrent structure in feature extraction and anti-noise. In Nnabuife et al. (2019), PCA was used to extract features and anti-noise for the ultrasonic signals, but the accuracy obtained by its classifier was only about 88%. Convolutions and stride-convolutions expand the receptive field while gradually extracting features in the signal. It is noteworthy that the information in the ultrasound signal can be roughly divided into "information valid for a specific task" and "information invalid for a specific task." Traditional artificial feature extraction methods (such as PCA or highpass filter) can only extract information in a fixed form, and task change can cause corresponding changes in the definition of "valid information." The convolution structures in Experiments No.1, 2, and 3 provide a more intelligent feature extraction approach than the traditional artificial features, which explains the excellent performance of each classifier in Section 5.1.1: "The classifiers depicted in Section 5.1.1 have good anti-noise performance". Furthermore, the RNN classifier in this study refers to the classifier that does not use any convolutional structures. The difference in Fig. 16 than in Section 5.1.1 illustrates the advantages of the above-mentioned convolutional structure in terms of anti-noise.

4.1.3. CRNN1.x-based flow regime classifiers

Figs. 17 and 25 show the results of the CRNN1.0-based classifier, which is the first combined classifier of convolutional and recurrent structures. Table 3 shows that the number of parameters of CRNN1.0 is reduced through the introduction of time-series features compared to the FCN classifier (Experiments No.5 and No.6 have decreased the complexity by 9.2% and 37.4%, respectively). The CRNN1.0 obtains 100%, 97.45%, and 97.63% of the training, testing, and validation accuracy, which shows that the flow regime classifier's performance achieves further improvement. Fig. 17 (a) shows that the convergence speed of CRNN1.0 has been accelerated, and the accuracy curve converges at about 25 epochs. The loss curves in Fig. 17 (b) and (c) are smoother

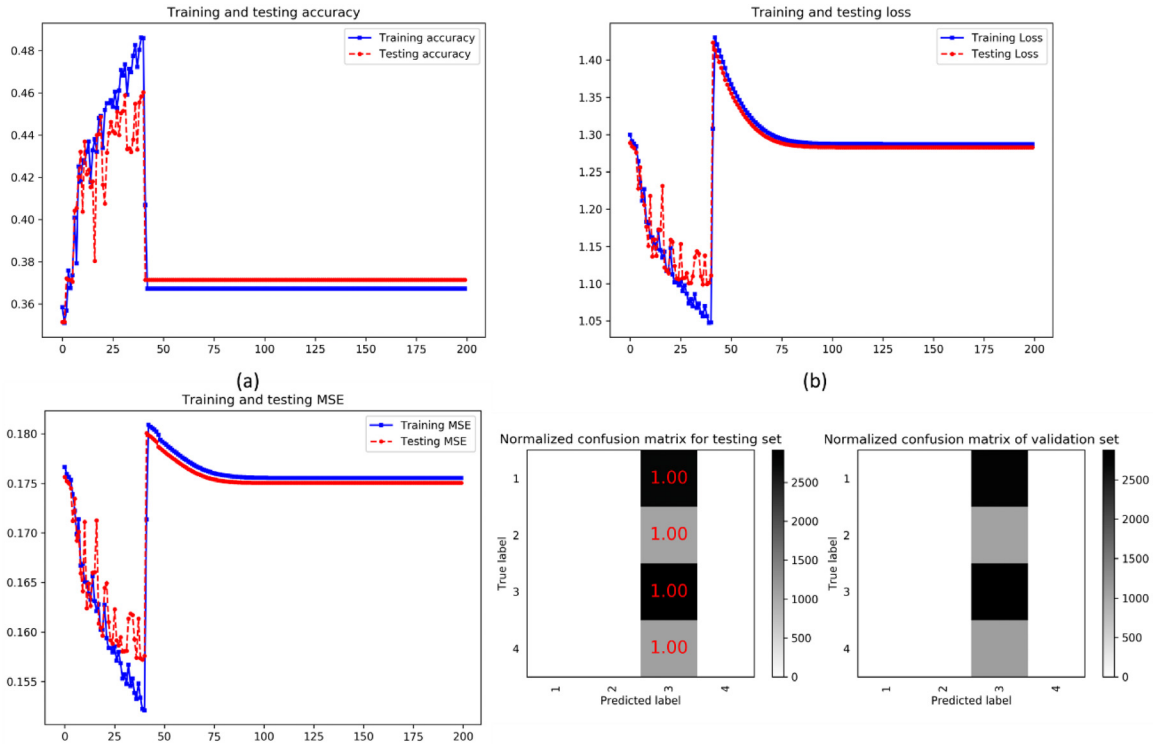


Fig. 16. The detailed experimental records of the RNN-based flow regime classifier using the augmentation scheme A (Experiment No.4 in Table 2 and Table 3)

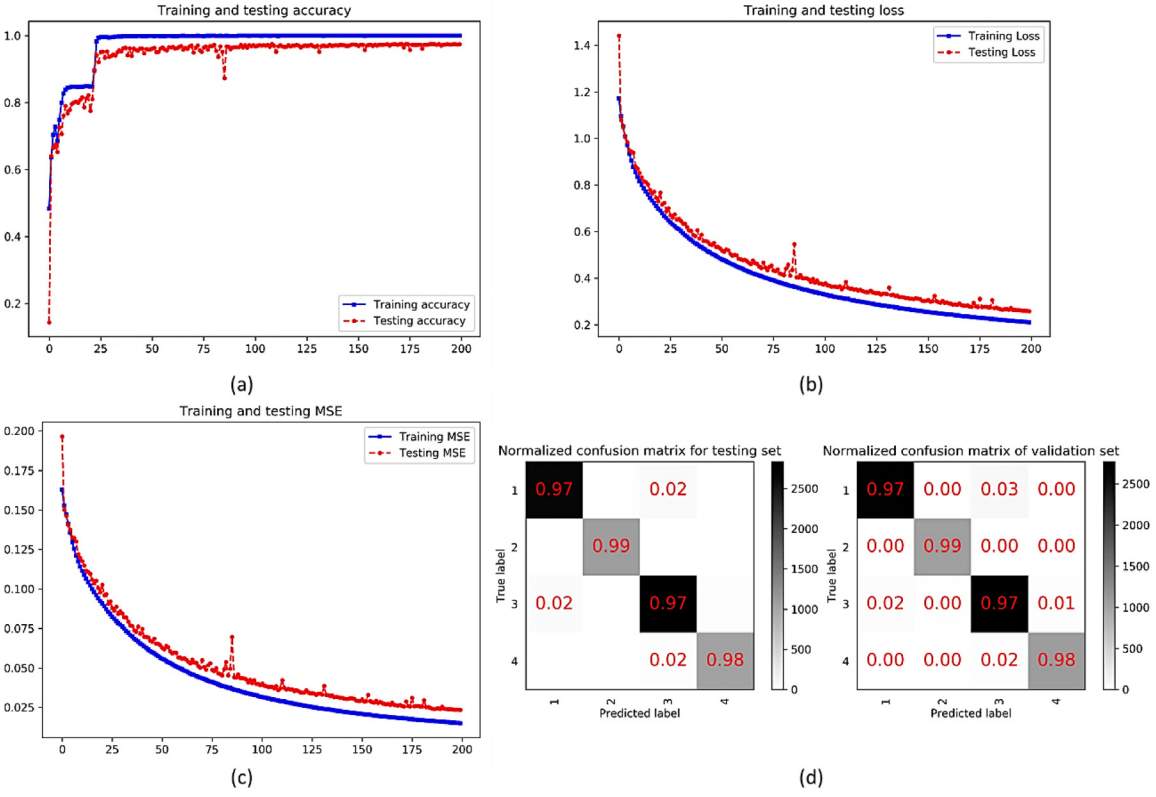


Fig. 17. The detailed experimental records of the CRNN1.0-based flow regime classifier using the augmentation scheme A (Experiment No.5)

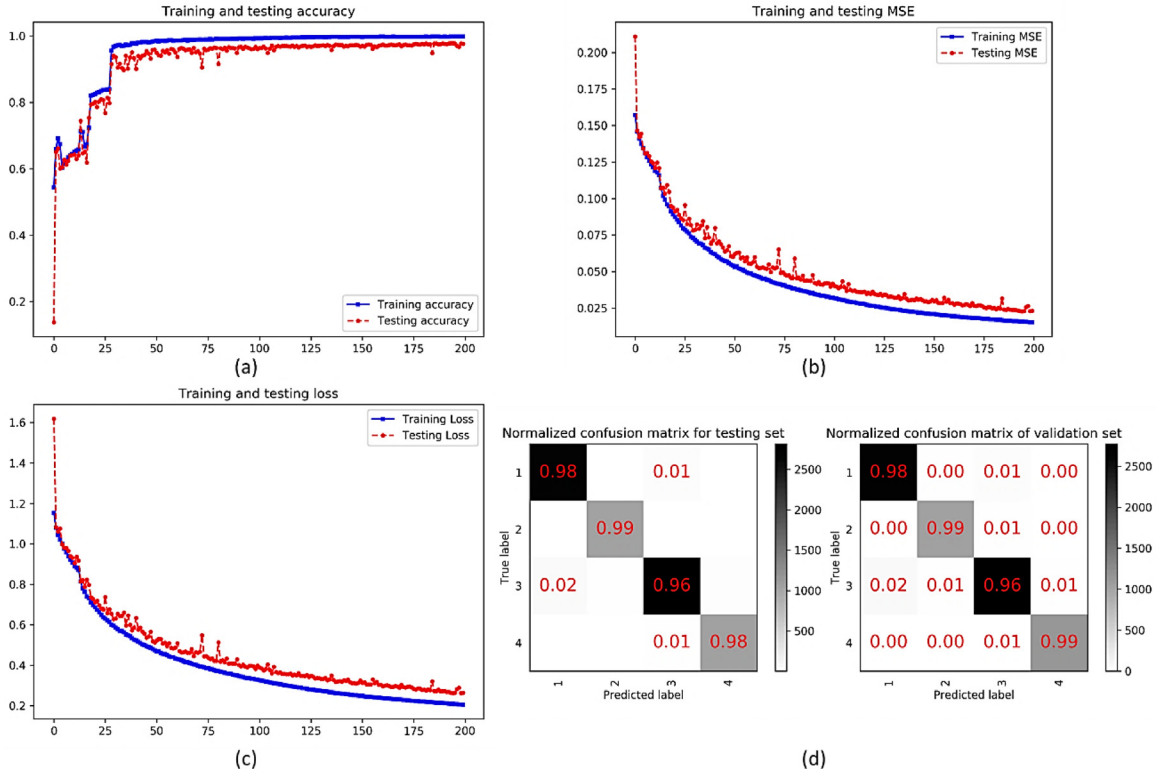


Fig. 18. The detailed experimental records of the CRNN2.1-based flow regime classifier using the augmentation scheme A in Table 2 (Experiment No.7)

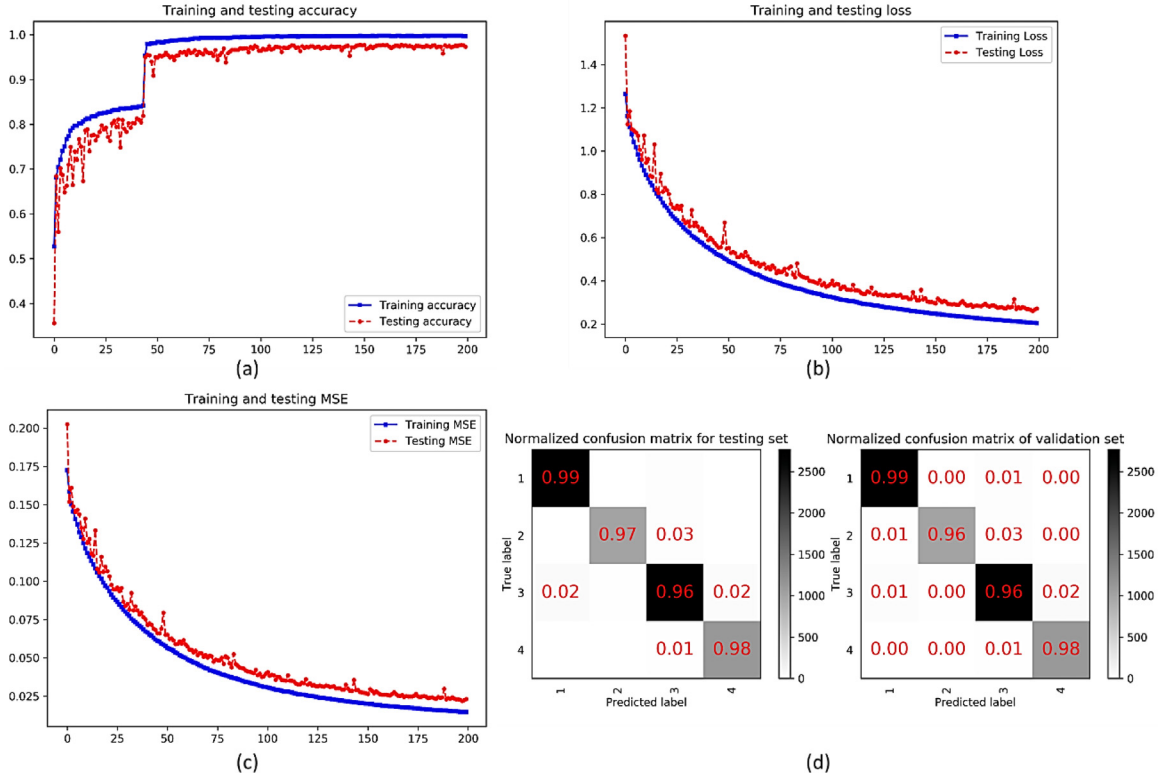


Fig. 19. The detailed experimental records of the CRNN2.2-based flow regime classifier using the augmentation scheme A in Table 2 (Experiment No.9)

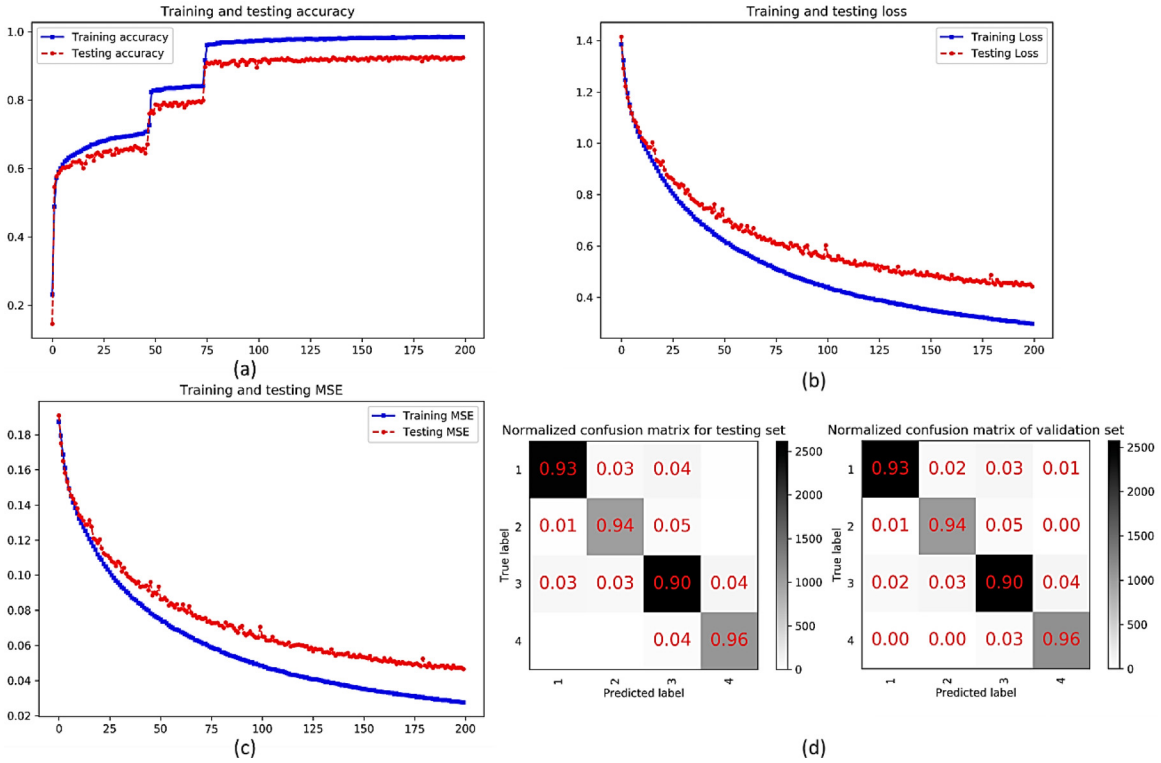


Fig. 20. The detailed experimental records of the CRNN3.0-based flow regime classifier using the augmentation scheme A (Experiment No.11)

than in the FCN classifier, indicating that the CRNN1.0 model is more stable. Notably, CRNN1.0 (Fig. 25) shows better performance than both the FCN classifier and (Nnabuife et al., 2019) when the augmentation scheme B (Table 1) is adopted.

4.1.4. CRNN2.x-based flow regime classifiers

Fig. 18 and Fig. 26 show the results of the CRNN2.1 classifier. The performance of CRNN2.1 on the augmentation scheme A performs well, and it has achieved the highest testing and validation accuracy of all fourteen experiments. Table 3 shows that the training, testing, and validation accuracy of CRNN2.1 obtained on the augmentation scheme A are 100%, 97.45%, and 98.06% respectively, which is higher performance than the state-of-the-art result (Nnabuife et al., 2020). Fig. 18 (d) left and right also show that the classification performance of the CRNN2.1 for the four flow regimes also achieves accurate results. However, the complexity of CRNN2.1 also reached an astonishing 17,113,606 parameters (Experiment No.7) and 14,242,542 (Experiment No.8), which increases about 130% and 91.5%, respectively compared to the FCN model. Therefore, although CRNN2.1 has improved accuracy by about 1.5% compared to FCN, its model has become very complex. However, when the $length_{input}$ is reduced to 8,192 data points (augmentation scheme B in Table 2), the training, testing, and validation accuracy significantly drop to below 40% (Fig. 26 (a), (b), and (c)). The reason for this abnormal situation is that the CRNN2.1 overfitted when the $length_{input}$ becomes shorter.

Figs. 19 and 27 (of CRNN2.2) maintain the same design layouts as CRNN2.1 but decrease the model complexity to verify the discussion in Experiment No.8. Experiment No.9 lower 20% of complexity than Experiment No.7 and Experiment No.10 lower 72% complexity than Experiment No.8. For the augmentation scheme A (Table 2), the training, testing, and verification accuracy in Experiment No.9 only dropped by about 0.2% to 0.8%, which is negligible compared to the reduction in model complexity. For the augmentation scheme B (Table 2), the accuracy has also been significantly improved, and its training, testing, and validation accuracy of Experiment No.10 are 85.81%, 76.26%, and

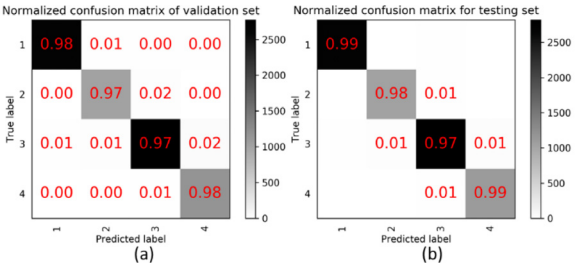


Fig. 21. The testing and validation confusion matrix of the converged CRNN3.1 flow regime classifier

100%, respectively. Although the accuracy in Fig. 27 (a) is closed to the level of [6], the testing and validation confusion matrix shows that CRNN2.1 has difficulty to distinguish the first and second types of flow regimes (Fig. 27 (d) left and right).

4.1.5. CRNN3.x-based flow regime classifiers

Figs 20 (Experiment No.11), 28 (Experiment No.12), and 29 (Experiment No.13) illustrate the performance of CRNN3.x among all three augmentation schemes. Although the training, testing, and validation accuracy of Experiment No.11 is 98.47%, 92.46%, and 93.06, the model complexity has significantly reduced by 98.4% (only contains 117,702 parameters). The extremely low complexity of CRNN3.x benefits from the design of the multi-scaled PST vector-set layout and the efficient usage of time-series features through CRNN. Notably, the loss and MSE curves in Fig. 20 (b) and (c) keep decreasing, indicating a good learning trend. Therefore, Experiment No.11 can achieve better results by extending the training time (see Section 5.2).

4.2. Further evaluation for the CRNN3.0-based flow regime classifier

This research further trained the CRNN3.0 model from Experiment No.11 to explore the convergence point. After another 100 epochs, the

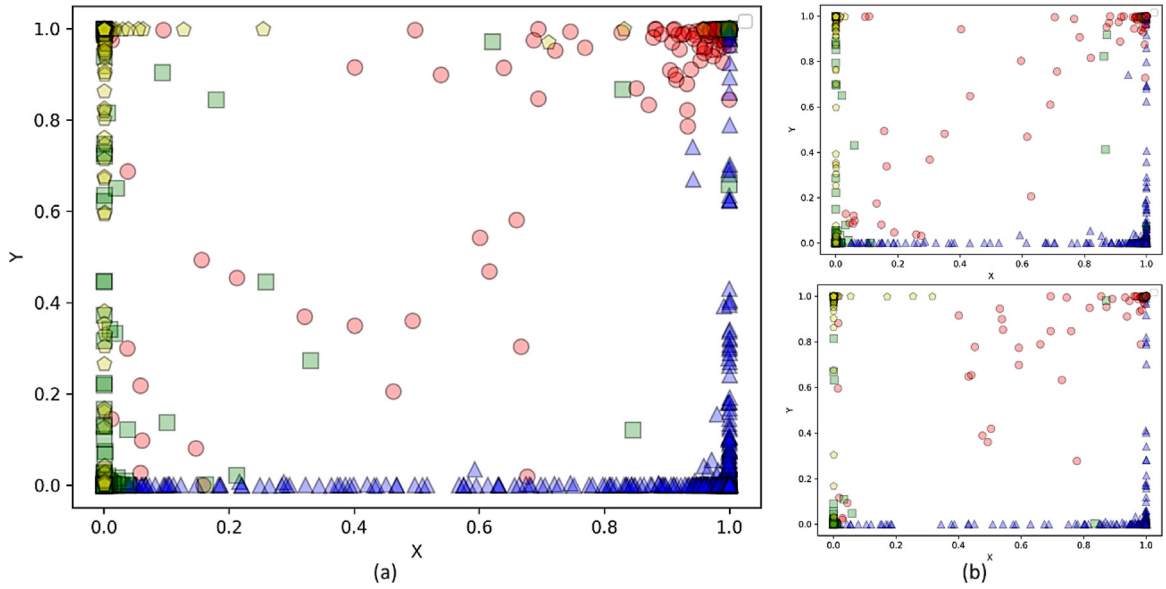


Fig. 22. Clusters visualization for the four flow regimes. The yellow pentagons, red circles, blue triangles, and green squares respectively correspond to the four flow regimes. (a), (b) top, and (b) bottom respectively indicate the clusters visualization data using 25%, 15%, and 10% of the data. x and y-axis respectively correspond to the first and second values in the 2DMap-sigmoid block, which is the output interface for all the classifiers.

training, testing, and validation accuracy arrived at 99.30%, 98.26%, and 97.80%, respectively. The training and testing categorical cross entropy reach 0.0865 and 0.1235, while the training and testing MSE separately corresponds to 0.0048 and 0.0092. Fig. 21 (a) and (b) separately depict the testing and validation confusion matrix. Although the converged CRNN3.0 does not defeat the best record in (Kuang et al., 2020), it does achieve a better result than the baseline from (Nnabuife et al., 2019) and the records from (Nnabuife et al., 2020), which verifies the discussion in Section 5.1.5.

The computation time for the CRNN-3.0-based flow regime classifier is less than 0.6 seconds for an implementation in Python on a standard desktop PC.

4.3. Clusters visualization for the four flow regimes

Fig. 22 shows the result of the 2DMap-sigmoid (mentioned in Section 4.1), a 2D cluster visualization of flow classification. Although Section 5.1 uses the accuracy curves, loss curves, MSE curves, testing set confusion matrix, and validation set confusion matrix to analyze and discuss the classifiers' results, it is still not explicit enough. This research provides a 2D mapping method for each classifier in the benchmark through the 2DMap-sigmoid interface, which uses the sigmoid function as the activation. Therefore, all flow ultrasonic signals can be mapped into a one by one square area. Fig. 22 is the 2D projection result of Experiment 5. The red circle represents the slugging flow; the yellow pentagon represents the bubbly flow; the blue triangle represents the churn flow; the green square represents the annular flow. The transparency of each sample in Fig. 22 is 75% so that the color can represent the number of overlapping targets in an area. For example, the four corners' colors are quite deep, indicating that the clustering orientations of the four flow regimes are the four corners. However, part of the red circles and green squares scatters throughout the 2D area explain that the identification accuracy of the first and third flow regimes is generally lower than the second and fourth flow regimes in the Section 5.1 benchmark. Fig. 22 provides a more explicit way to visualize results qualitatively.

5. Conclusion

This research proposed an ultrasonic sensor and a convolutional recurrent neural network (CRNN) for gas-liquid two-phase flow recognition in an S-shaped riser, achieving testing and validation accuracies of 98.13 % and 98.06 %, respectively, while reducing complexity by 98.4 % (only 117,702 parameters).

Compared with the previous methods, the applied time-domain characteristics in this research successfully improved the identification accuracy and simplified the deep learning model complexity. This research discussed the promising ability of CNN-RNN cooperation for the ultrasonic signal for flow regime identification in S-shape risers (the theoretical logics in Section 4 and the experimental validations in Section 5). Notably, this research also proposes an explicit 2D clustering visualization method, which qualitatively illustrates the performance of CRNN-based flow regime identification.

This contribution proposed an end-to-end (E2E) CRNN-based solution towards the gas-liquid flow regime identification using ultrasonic signals in an S-shaped riser. The proposed CRNN classifier significantly decreases the module complexity along with a high identification accuracy. The cooperation between the feature extraction of CNN and time-domain analysis of RNN shows considerable ability for handling the low signal-noise ratio and long ultrasonic signal. The systematically investigated CRNN schemes provide a benchmark (or baseline) for further related research (e.g. transfer learning).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank the Process Systems Engineering Group at Cranfield University for advice and support. Furthermore, this work was supported by grants from the National Natural Science Foundation of China (No. 51706224), the Tianjin Municipal Education Commission (2019KJ120), and the Tianjin Municipal Natural Science Foundation (17JCQNJC04600).

Appendix

Experimental records for the CRNN-based flow regime identification benchmark

Fig. 23-29

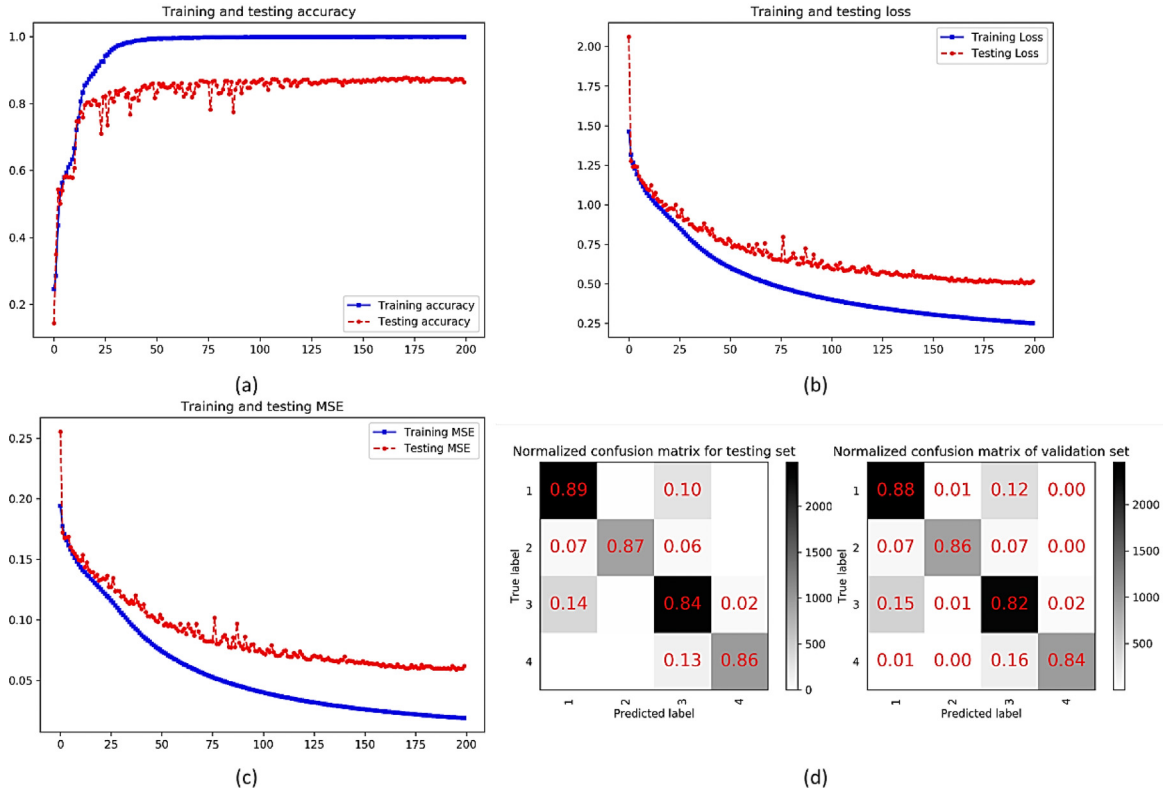


Fig. 23. The detailed experimental records of the FCN-based flow regime classifier using the augmentation scheme B (Experiment No.2).

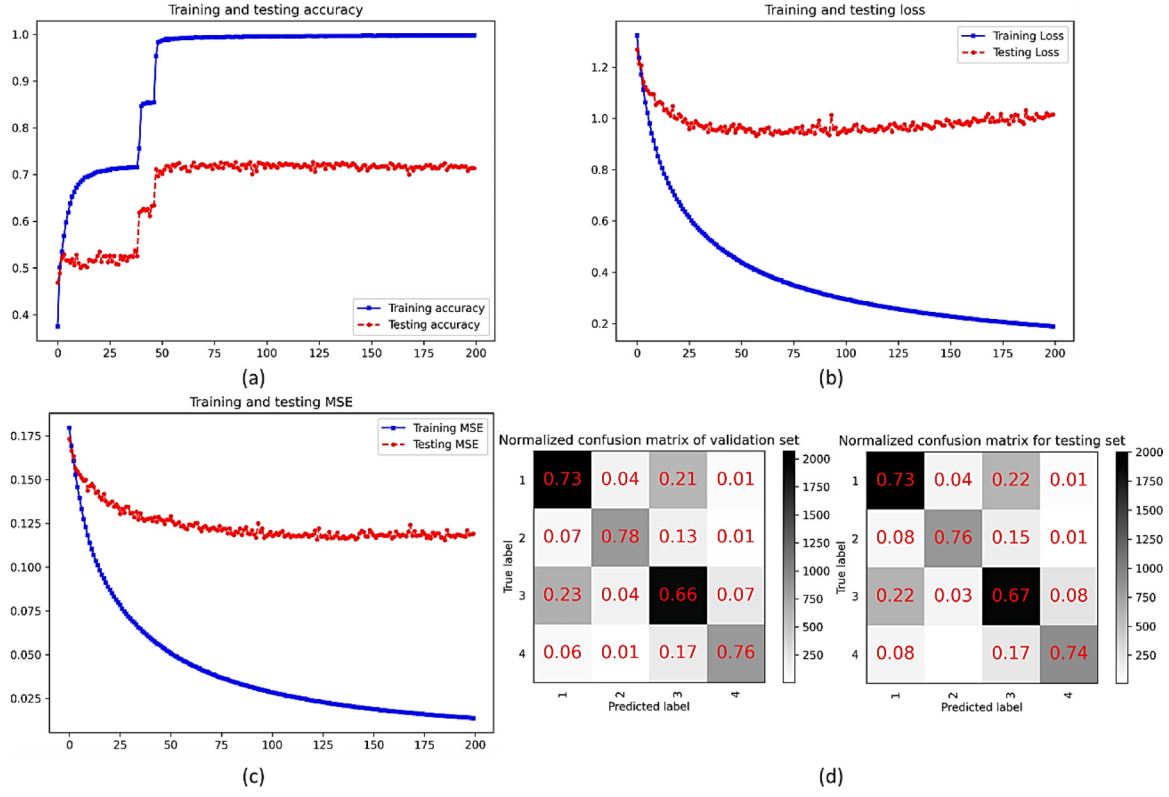


Fig. 24. The detailed experimental records of the FCN-based flow regime classifier using the augmentation scheme C (Experiment No.3).

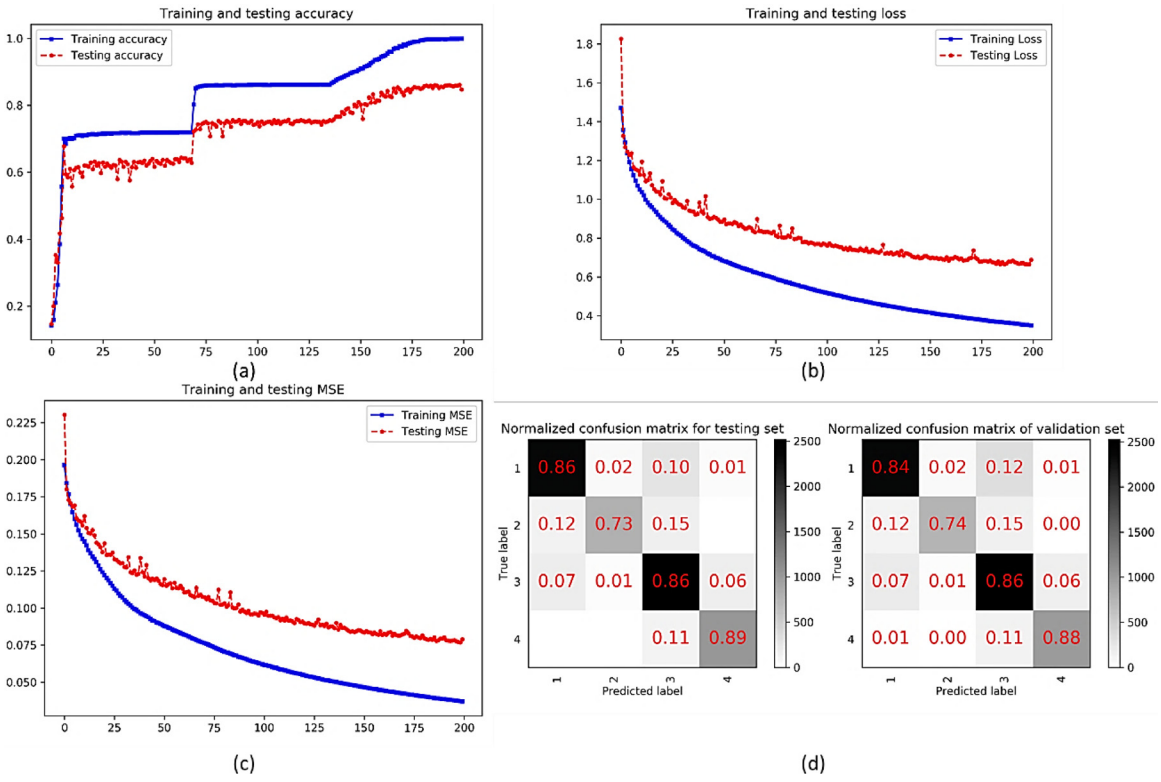


Fig. 25. The detailed experimental records of the CRNN1.0-based flow regime classifier using the augmentation scheme B (Experiment No.6).

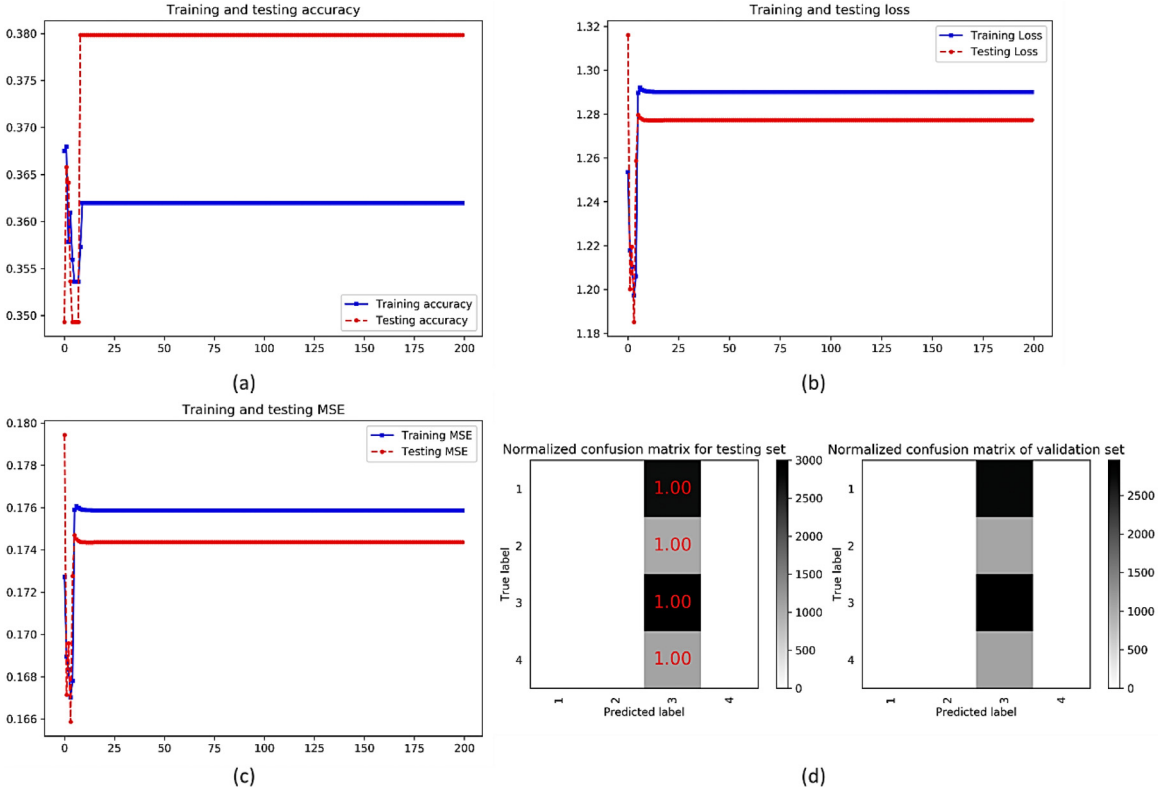


Fig. 26. The detailed experimental records of the CRNN2.1-based flow regime classifier using the augmentation scheme B (Experiment No.8).

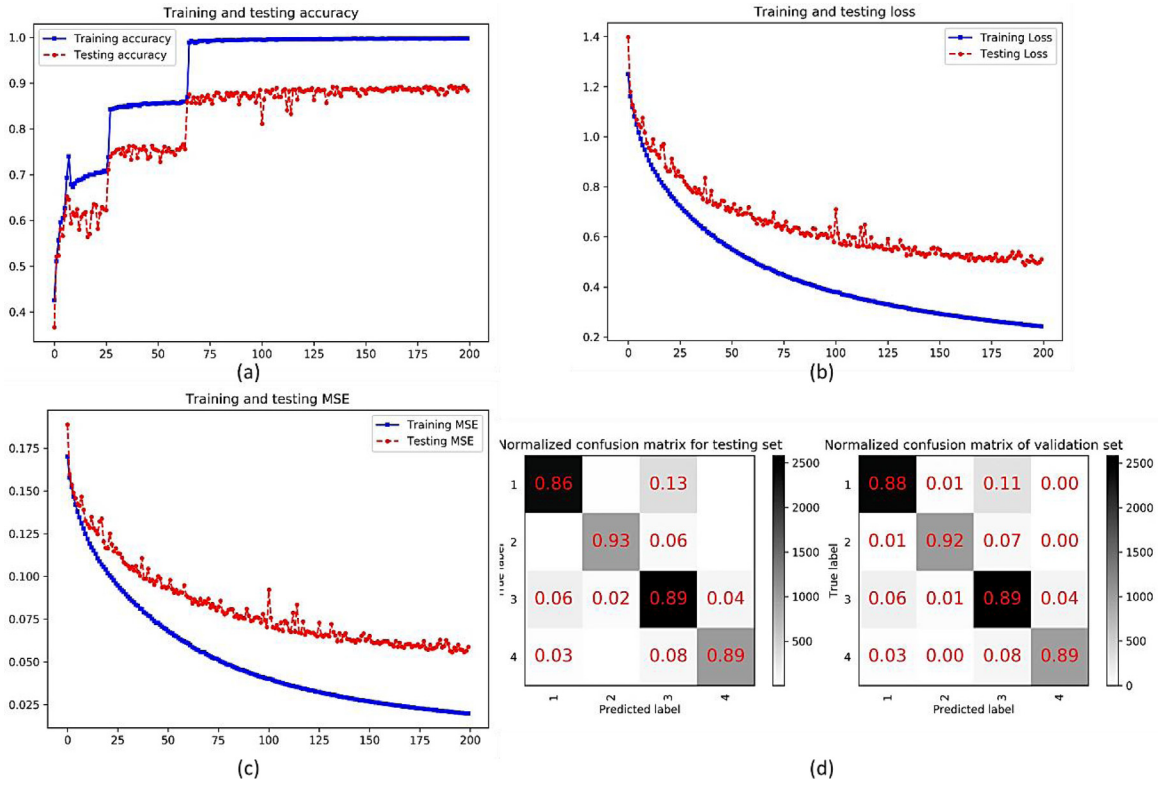


Fig. 27. The detailed experimental records of the CRNN2.2-based flow regime classifier using the augmentation scheme B (Experiment No.10).

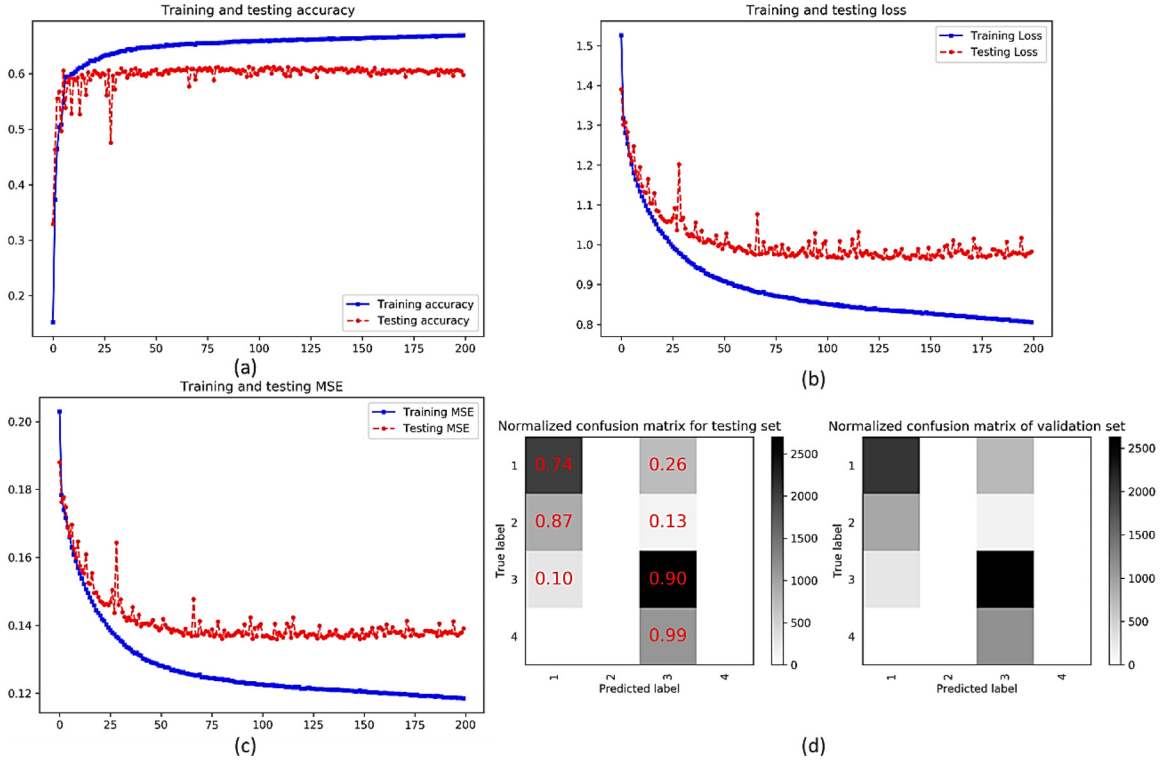


Fig. 28. The detailed experimental records of the CRNN3.0-based flow regime classifier using the augmentation scheme B (Experiment No.12).

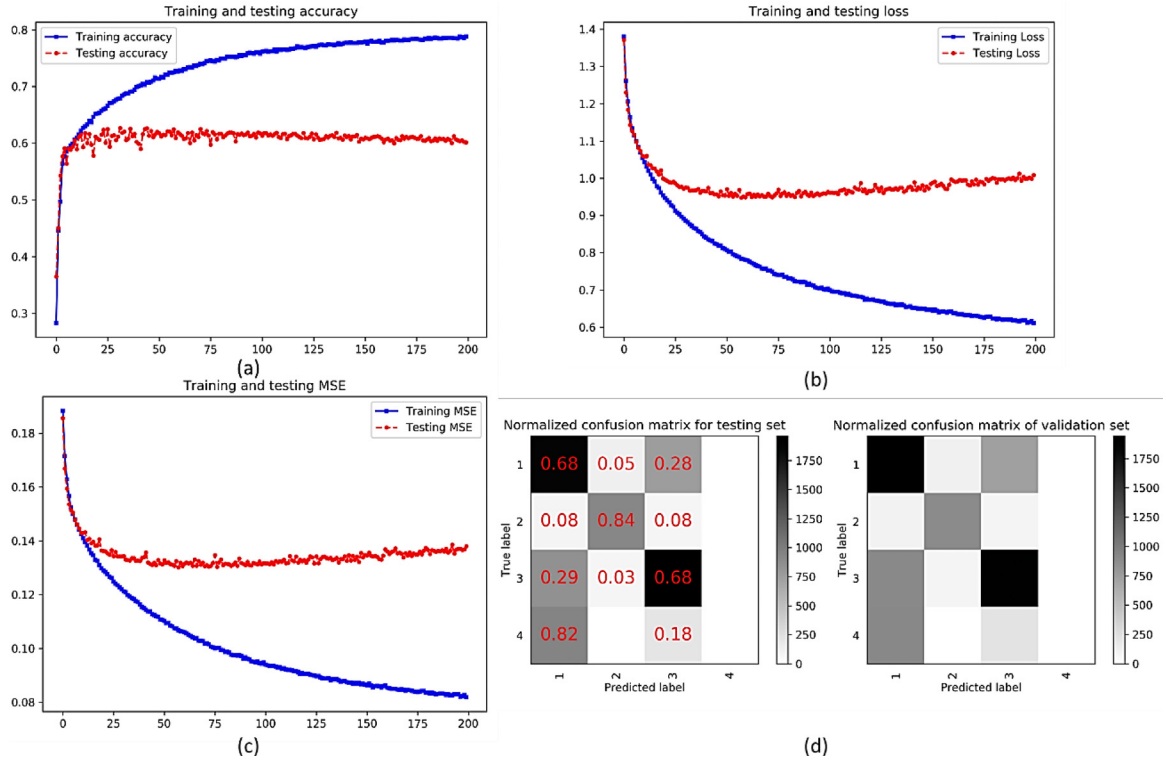


Fig. 29. The detailed experimental records of the CRNN3.0-based flow regime classifier using the augmentation scheme C (Experiment No.13).

References

Affonso, R.R.W., et al., 2020. Flow regime and volume fraction identification using nuclear techniques, artificial neural networks and computational fluid dynamics. Appl. Radiat. Isot. 159 (May 2019), 109103. doi:10.1016/j.apradiso.2020.109103.

Chakraborty, S., Das, P.K., 2018. A unique methodology of objective regime classification for two phase flow based on the intensity of digital images. Exp. Therm Fluid Sci. 99 (July), 537–546. doi:10.1016/j.expthermflusci.2018.07.037.

Chalgeri, V.S., Jeong, J.H., 2019. Flow regime identification and classification based on void fraction and differential pressure of vertical two-phase flow in rectangular channel. Int. J. Heat Mass Transfer 132, 802–816. doi:10.1016/j.ijheatmasstransfer.2018.12.015.

Cobbold, R., 1989. Doppler ultrasound: physics, instrumentation, and clinical applications. J. Biomed. Eng. 11, 528. doi:10.1016/0141-5425(89)90051-4.

Deng, M., et al., 2020. Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. Neural Netw. 130, 22–32. doi:10.1016/j.neunet.2020.06.015.

Dong, X., et al., 2015. Oil-water two-phase flow velocity measurement with continuous wave ultrasound Doppler. Chem. Eng. Sci. 135 (2), 155–165. doi:10.1146/annurev.fl.24.010192.000321.

Falcone, G., et al., 2018. Assessment of deep geothermal energy exploitation methods: The need for novel single-well solutions. Energy 160, 54–63. doi:10.1016/j.energy.2018.06.144.

Fang, L., et al., 2020. Identification of two-phase flow regime using ultrasonic phased array. Flow Meas. Instrum. 72 (September 2019), 101726. doi:10.1016/j.flowmeasinst.2020.101726.

Feng, Z., et al., 2021. Experimental investigation on gas-liquid two-phase flow distribution characteristics in parallel multiple channels. Exp. Therm Fluid Sci. 127 (November 2020), 110415. doi:10.1016/j.expthermflusci.2021.110415.

Figueiredo, M., de, M.F., et al., 2020. Flow pattern classification in water-air vertical flows using a single ultrasonic transducer. Exp. Therm Fluid Sci. 119 (January). doi:10.1016/j.expthermflusci.2020.110189.

Figueiredo, M.M.F., et al., 2016. 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. Exp. Therm Fluid Sci. 70, 29–50. doi:10.1016/j.expthermflusci.2015.08.010.

George, D.L., et al., 2000. Validation of electrical-impedance tomography for measurements of material distribution in two-phase flows. Int. J. Multiphase Flow 26 (4), 549–581. doi:10.1016/S0301-9322(99)00029-4.

Ghavamian, F., Simone, A., 2019. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Comput. Meth. Appl. Mech. Eng. 357, 112594. doi:10.1016/j.cma.2019.112594.

Hammad, F.A., et al., 2021. Internal two-phase flow and spray characteristics of outside-in-liquid twin-fluid atomizers. Appl. Therm. Eng. 187 (January), 116555. doi:10.1016/j.applthermaleng.2021.116555.

Hanus, R., 2015. Application of the Hilbert Transform to measurements of liquid-gas flow using gamma ray densitometry. Int. J. Multiphase Flow 72, 210–217. doi:10.1016/j.ijmultiphaseflow.2015.02.002.

Hanus, R., et al., 2018. Identification of liquid-gas flow regime in a pipeline using gamma-ray absorption technique and computational intelligence methods. Flow Meas. Instrum. 60 (September 2017), 17–23. doi:10.1016/j.flowmeasinst.2018.02.008.

He, K. et al. (2015).'Deep Residual Learning for Image Recognition', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, W., et al., 2020. Comparing SNNs and RNNs on neuromorphic vision datasets: similarities and differences. Neural Netw. 132, 108–120. doi:10.1016/j.neunet.2020.08.001.

Jeon, S., Moon, J., 2020. Malware-detection method with a convolutional recurrent neural network using opcode sequences. Inf. Sci. 535, 1–15. doi:10.1016/j.ins.2020.05.026.

Kou, J., Sun, S., 2018. Entropy stable modeling of non-isothermal multi-component diffuse-interface two-phase flows with realistic equations of state. Comput. Meth. Appl. Mech. Eng. 341, 221–248. doi:10.1016/j.cma.2018.06.002.

Kuang, B., Nnabuife, S.G., Rana, Z., 2021. Pseudo-image-feature-based identification benchmark for multi-phase flow regimes. Chem. Eng. J. Adv. 5 (November 2020), 100060. doi:10.1016/j.ceja.2020.100060.

Kuang, B., Nnabuife, S.G., Rana, Z., 2020. Pseudo-image-feature-based identification benchmark for multi-phase flow regimes. Chem. Eng. J. Adv., 100060 doi:10.1016/j.ceja.2020.100060.

Li, N., Guo, L., Li, W., 2013. Gas-liquid two-phase flow patterns in a pipeline-riser system with an S-shaped riser. Int. J. Multiphase Flow 55, 1–10. doi:10.1016/j.ijmultiphaseflow.2013.04.003.

Li, W., Guo, L., Xie, X., 2017. Effects of a long pipeline on severe slugging in an S-shaped riser. Chem. Eng. Sci. 171, 379–390. doi:10.1016/j.ces.2017.05.017.

Lin, Z., et al., 2020. Prediction of two-phase flow patterns in upward inclined pipes via deep learning. Energy 210, 118541. doi:10.1016/j.energy.2020.118541.

Liu, X., Falcone, G., Alimonti, C., 2018. A systematic study of harnessing low-temperature geothermal energy from oil and gas reservoirs. Energy 142, 346–355. doi:10.1016/j.energy.2017.10.058.

Mao, N., et al., 2022. Black-box real-time identification of sub-regime of gas-liquid flow using Ultrasound Doppler Velocimetry with deep learning. Energy 239, 122319. doi:10.1016/j.energy.2021.122319.

Murai, Y., et al., 2010. Ultrasonic detection of moving interfaces in gas-liquid two-phase flow. Flow Meas. Instrum. 21 (3), 356–366. doi:10.1016/j.flowmeasinst.2010.03.007.

Nnabuife, S.G., Pilario, K.E.S., et al., 2019. Identification of gas-liquid flow regimes using a non-intrusive Doppler ultrasonic sensor and virtual flow regime maps. Flow Meas. Instrum. 68 (October 2018), 101568. doi:10.1016/j.flowmeasinst.2019.05.002.

Nnabuife, S.G., et al., 2020. Non-Intrusive classification of gas-liquid flow regimes in an s-shaped pipeline riser using a Doppler ultrasonic sensor and deep neural networks. Chem. Eng. J. 403, 126401. doi:10.1016/j.cej.2020.126401.

Nnabuife, S.G., Kuang, B., et al., 2021. Development of gas-liquid flow regimes identification using a noninvasive ultrasonic sensor, belt-shape features, and convolutional neural network in an s-shaped riser. In: IEEE Transactions on Cybernetics, pp. 1–15. doi:10.1109/TCYB.2021.3084860.

Nnabuife, S.G., Sharma, P., et al., 2021. Development of gas-liquid slug flow measurement using continuous-wave doppler ultrasound and bandpass power spectral density. ChemEngineering 5 (1), 1–14. doi:10.3390/chemengineering5010002.

Nnabuife, S.G., Whidborne, J., Lao, L., 2019. *Two-phase gas-liquid flow regimes identification in an s-shape pipeline-riser using doppler ultrasonic sensor, Cranfield Online Research Data (CORD) Repository*. Cranfield: Cranfield Online Research Data (CORD) Repository doi:10.17862/cranfield.rd.11369379.

O'Shea, K., and Nash, R. (2015). 'An Introduction to Convolutional Neural Networks', pp. 1–11.

Nnabuife, S.G., Whidborne, J., Lao, L., Cao, Y., 2019a. Venturi multiphase flow measurement based active slug control. ICAC 2019 - 2019 25th IEEE International Conference on Automation and Computing doi:10.23919/IConAC.2019.8895212.

Parrales, A., et al., 2018. New void fraction equations for two-phase flow in helical heat exchangers using artificial neural networks. Appl. Therm. Eng. 130, 149–160. doi:10.1016/j.applthermaleng.2017.10.139.

Pascanu, R., et al., 2013. How to Construct Deep Recurrent Neural Networks. In: 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, pp. 1–13.

Qin, C., et al., 2019. Convolutional recurrent neural networks for dynamic mr image reconstruction. IEEE Trans. Med. Imaging 38 (1), 280–290. doi:10.1109/TMI.2018.2863670.

Queiroz, L.H., et al., 2021. Physics-informed deep learning to predict flow fields in cyclone separators. Digital Chem. Eng., 100002 doi:10.1016/j.dche.2021.100002.

Rosa, E.S., et al., 2010. Performance comparison of artificial neural networks and expert systems applied to flow pattern identification in vertical ascendant gas-liquid flows. Int. J. Multiphase Flow 36 (9), 738–754. doi:10.1016/j.ijmultiphaseflow.2010.05.001.

Saltzer, J.H., Reed, D.P., Clark, D.D., 1984. End-to-end arguments in system design. ACM Trans. Comp. Syst. 2 (4), 277–288. doi:10.1145/357401.357402.

Shen, X., Hibiki, T., 2021. Distribution parameter and drift velocity for upward gas-liquid metal two-phase flow. Appl. Therm. Eng. 184 (October 2020), 116242. doi:10.1016/j.applthermaleng.2020.116242.

Sunde, C., Avdic, S., Pázsit, I., 2005. Classification of two-phase flow regimes via image analysis and a neuro-wavelet approach. Prog. Nucl. Energy 46 (3–4), 348–358. doi:10.1016/j.pnucene.2005.03.015.

Tan, C., et al., 2021. Ultrasonic doppler technique for application to multiphase flows: a review. Int. J. Multiphase Flow, 103811 doi:10.1016/j.ijmultiphaseflow.2021.103811, Pergamon.

Thorn, R., Johansen, G.A., Hjertaker, B.T., 2013. Three-phase flow measurement in the petroleum industry. Meas. Sci. Technol. 24 (1), 17. doi:10.1088/0957-0233/24/1/012003.

Wada, S., Kikura, H., Aritomi, M., 2006. Pattern recognition and signal processing of ultrasonic echo signal on two-phase flow. Flow Meas. Instrum. 17 (4), 207–224. doi:10.1016/j.flowmeasinst.2005.11.006.

Wiedemann, P., et al., 2019. Fuzzy flow pattern identification in horizontal air-water two-phase flow based on wire-mesh sensor data. Int. J. Multiphase Flow 117, 153–162. doi:10.1016/j.ijmultiphaseflow.2019.05.004.

Xu, Q., Li, W., et al., 2020. Intelligent recognition of severe slugging in a long-distance pipeline-riser system. Exp. Therm Fluid Sci. 113 (August 2019), 110022. doi:10.1016/j.expthermflusci.2019.110022.

Xu, Q., Zhou, H., et al., 2020. Study of identification of global flow regime in a long pipeline transportation system. Powder Technol. 362, 507–516. doi:10.1016/j.powtec.2019.12.018.

Yao, H., Gao, Y., Liu, Y., 2020. FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction. Comput. Meth. Appl. Mech. Eng. 363, 112892. doi:10.1016/j.cma.2020.112892.

Yin, P., et al., 2020. Investigation of slug flow characteristics in hilly terrain pipeline using ultrasonic Doppler method. Chem. Eng. Sci. 211, 115300. doi:10.1016/j.ces.2019.115300.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). 'Recurrent Neural Network Regularization', (2013)., pp. 1–8.

Zhai, S., et al., 2016. Doubly convolutional neural networks. Mach. Learn. 2 (1), 173–191. doi:10.1016/B978-0-12-815739-8.00010-9.

Zhang, Zhao, et al., 2021. Dense residual network: enhancing global dense feature flow for character recognition. Neural Netw. 139, 77–85. doi:10.1016/j.neunet.2021.02.005.



https://dspace.org/

Gas-liquid flow regimes identification using non-intrusive Doppler ultrasonic sensor and convolutional recurrent neural networks in an S-shaped riser

Kuang, Boyu; Nnabuife, Somtochukwu Godfrey; Sun, Shuang; Whidborne, James F.; Rana, Zeeshan A.

2022-01-19
Elsevier
https://doi.org/10.1016/j.dche.2022.100012
Kuang B, Nnabuife SG, Sun S, et al., (2022) Gas-liquid flow regimes identification using non-intrusive Doppler ultrasonic sensor and convolutional recurrent neural networks in an S-shaped riser, Digital Chemical Engineering, Volume 2, March 2022, Article number 100012

Downloaded from a DSpace repository. https://dspace.org/