






RESEARCH ARTICLE | OCTOBER 03 2023

A machine learning study to predict wind-driven water runback characteristics

Jincheng Wang ; Haiyang Hu ; Ping He ; Hui Hu  



Physics of Fluids 35, 102104 (2023)

<https://doi.org/10.1063/5.0167545>



View
Online



Export
Citation

CrossMark

Articles You May Be Interested In

Completing the dark matter solutions in degenerate Kaluza-Klein theory

J. Math. Phys. (April 2019)

Gibbs measures based on 1d (an)harmonic oscillators as mean-field limits

J. Math. Phys. (April 2018)

An upper diameter bound for compact Ricci solitons with application to the Hitchin–Thorpe inequality. II

J. Math. Phys. (April 2018)

A machine learning study to predict wind-driven water runback characteristics

Cite as: Phys. Fluids **35**, 102104 (2023); doi: [10.1063/5.0167545](https://doi.org/10.1063/5.0167545)

Submitted: 13 July 2023 · Accepted: 15 September 2023 ·

Published Online: 3 October 2023



View Online



Export Citation



CrossMark

Jincheng Wang, Haiyang Hu, Ping He, and Hui Hu^{a)}

AFFILIATIONS

Department of Aerospace Engineering, Iowa State University, Ames, Iowa 50011-1096, USA

^{a)} Author to whom correspondence should be addressed: huhui@iastate.edu

ABSTRACT

The unsteady runback behavior of wind-driven runback water film (WDRWF) flows over aircraft surfaces has a significant impact on the aircraft icing process, one of the most significant aviation hazards in cold weather. The limited understanding of the complex multiphase interactions between freestream airflow, water film motion, and solid airframe surface makes conventional theoretical/numerical methods unable to precisely simulate WDRWF flow. Machine learning-based techniques can accurately capture complex physics using data, making it an attractive alternative to conventional methods. In this study, machine learning methods are used to predict the evolution of the front contact point (FCP) of WDRWF flow and film thickness distribution (FTD) of WDRWF flow. For FCP prediction, the performance of the Light Gradient-Boosting Machine (LightGBM) and Multi-Layer Perceptron is compared quantitatively. They perform well in capturing intermittent and smooth features, respectively. For the prediction of the spatial-temporal evolution of FTD, a computationally efficient deep neural network architecture named ConvLSTM-AutoEncoder was developed, which predicts a future FTD based on a sequence of FTDs in the past. The robustness of the ConvLSTM-AutoEncoder model to noisy input FTD is demonstrated. The generalizability of the three models is evaluated by applying the trained models to unexplored datasets. Based on the proposed techniques' generalizability, robustness, and computational efficiency, machine learning-based methods are demonstrated to be powerful tools in predicting the complex unsteady characteristics of the multiphase WDRWF flows.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0167545>

I. INTRODUCTION

Aircraft icing is a well-known weather hazard affecting aircraft performance and flight safety greatly. As ice structures accrete over airframe surfaces, the aerodynamic performance of an airplane would degrade significantly with the aerodynamic drag increasing dramatically and lift force decreasing rapidly. Ice accretion can usually be categorized into rime and glaze icing. When the ambient temperature is low enough (e.g., less than -8°C) under a relatively dry condition, the supercooled water droplets would be frozen instantly to form rime ice upon the impingement onto the airframe surface.^{1,2} Glaze icing usually occurs under conditions with relatively warmer ambient temperatures (i.e., just below the freezing point of water), higher liquid water content (LWC), and larger supercooled water droplets. In a glaze ice accretion process, only a portion of the supercooled water droplet would be frozen into solid ice upon impact with the airframe surfaces of an airplane, while the rest of the impacted water droplets would stay in liquid and move freely over the airframe surfaces in the form of water film/rivulet flows as driven by the boundary layer airflow.³⁻⁵ The wind-driven runback characteristics of the unfrozen water film/rivulet

flows would directly or indirectly affect the ice formation and accretion over the airframe surfaces.⁶ The transient behavior of unfrozen runback water flow would directly affect the transportation process of impinged water mass over the ice-accreting airframe surfaces.⁷ Local convective heat transfer and the rate of ice accumulation would be affected by the surface roughness, thereby affecting the characteristics of ice accumulation greatly.⁸⁻¹¹ While Waldman and Hu¹² conducted an experimental investigation to reveal clearly that the unsteady runback process of wind-driven runback water film (WDRWF) flows over airfoil surfaces would influence the ice accretion characteristics on aircraft airfoils/wings greatly, Hansman and Turnock¹³ also demonstrated that wind-driven runback of the unfrozen water flow over ice accreting airframe surface would modify the morphology of accreted glaze ice structures significantly.

Given the abundance of experimental data and challenges associated with aircraft icing, machine learning (ML) integration presents a promising opportunity to improve the understanding of the underlying pertinent to aircraft icing phenomena. Recently, ML techniques have been used widely in various research fields, including medical

diagnosis, image recognition, self-driving vehicles, and product recommendation. Neural networks are a subset of machine learning that form the basis of deep learning techniques. Their name and structure are derived from the human brain, emulating how organic neurons communicate. Several studies relevant to ice shapes and icing severities have been conducted to explore the potential of using neural networks to study aircraft icing phenomena. For example, Cao *et al.*¹⁴ utilized a neural network model to predict the impact of an ice horn's geometric parameters (e.g., position, height, diameter, etc.) on the aerodynamic performance of an airfoil model. McCann¹⁵ constructed a neural network to estimate icing severity based on the cloud liquid water content and the droplet size. Ogrretim *et al.*¹⁶ utilized a neural network model to predict ice formation on airfoils based on flight conditions. Yu *et al.*¹⁷ proposed a multi-autoencoder fusion network to fast predict ice profiles with icing conditions. Yang *et al.*¹⁸ integrated a backpropagation neural network with a K-MEANS clustering-convolutional neural network to predict droplet impingement and ice patterns on a supercooled surface. As mentioned before, the unfrozen runback water film significantly influences the shape of ice formed on the aircraft. On the surface of runback water film, surface waves are formed by aerodynamic shear force. Investigating the characteristics of surface waves would improve the understanding of the runback water transportation process and the ice formation process. Several machine learning-based studies have been conducted to predict the characteristics of surface waves. Deo and Jaiman¹⁹ developed an attention-based convolutional neural network based on numerical simulation data to predict shallow water wave propagation. Kagemoto²⁰ predicted the water-surface wave train's peak heights using a neural network based on field measurements in ocean waves. Sun *et al.*²¹ constructed a modified U-Net to predict the ocean surface wave's evolution.

In addition to neural networks, gradient boosting is another powerful ML method for regression and classification applications. It provides a prediction model as an ensemble of weak prediction models, often decision trees. In most cases, it outperforms the random forest algorithm.²² Extreme gradient boosting (XGBoost)²⁰ is an optimized gradient boosting framework renowned for its superior accuracy and efficiency.²³ LightGBM is an improved version of XGBoost. It is more efficient and accurate than the XGBoost model's disadvantage of scanning all data points while determining the greatest information gain.²⁴ Several studies^{25–27} have been conducted to successfully predict icing severity from flight conditions based on gradient boosting algorithms.

Considering the prevalence of time-dependent characteristics in various fluid dynamics problems, such as the flow of a wind-driven runback water film, the long short-term memory (LSTM) network emerges as a suitable machine learning (ML) tool for predicting time-dependent fluid dynamics phenomena. It is a subset of recurrent neural networks (RNNs) that can learn long-term dependencies, particularly in sequence prediction tasks.²⁸ Up to the date of writing this article, no study has been found to use the LSTM-based method to predict the runback water film evolution or ice accretion process related to the aircraft icing phenomenon. However, some successful applications of the LSTM-based method in predicting complex spatial-temporal fluid flow have been listed here to show the effectiveness of the LSTM-based method. Mohan and Gaitonde,²⁹ Pawar *et al.*,³⁰ and Deng *et al.*³¹ created models by combining the proper orthogonal decomposition (POD) and LSTM techniques to estimate the temporal characteristics of unsteady flows. Zhu *et al.*³² utilized a

convolutional long short-term memory (ConvLSTM) for predicting the spatial-temporal characteristics of turbulent flow dynamics. Li *et al.*³³ constructed a ConvLSTM network and utilized upstream flow data to forecast the downstream channel's supersonic flow. Han *et al.*³⁴ developed a hybrid neural network based on ConvLSTM to predict the periodic characteristics of cylinder and airfoil flows. Mi *et al.*³⁵ created a generator-discriminator network based on the concept of energy cascade to predict the cylinder wake velocity fields. Laima *et al.*³⁶ created a deep neural network to reconstruct spatiotemporal cylinder flow by combining a convolutional neural network with an RNN. Jin *et al.*³⁷ combined POD and RNN to reconstruct time-resolved cylinder flow, outperforming the extended POD method.

Despite the progress described above, most of the previous research evaluated the generalizability of the machine learning model by applying the trained model to unknown data that differs only slightly from the training set. In addition, previous research primarily employed the ConvLSTM model to predict flows with periodic characteristics, representing only a small portion of the actual flow phenomenon. Moreover, most researchers trained machine learning models with numerical data. It is difficult to apply this procedure when there are no reliable numerical or analytical solutions for certain complex flow problems, such as the wind-driven water film flow problem.

In the present study, we report the progress made in breaking the above limitations by developing effective machine learning methods that can (1) predict the characteristics of unsteady multiphase flows (i.e., WDRWF flows) that are significantly different from training data, (2) handle general, non-periodic unsteady multiphase flow phenomena, and (3) leverage experimental data (with noises) as inputs. The work presented here was motivated by the fact that minor changes in flow conditions (for example, incoming wind speed or water film flow rate) can cause significant variations in film flow properties such as film thickness distribution (FTD) and front contact point (FCP) development of WDRWF flows. The limited understanding of such complex multiphase interactions makes conventional theoretical/numerical methods unable to precisely simulate WDRWF flows. However, machine learning-based techniques can accurately capture complex physics using data, making it an attractive alternative to conventional methods. The current work focuses on developing machine learning methods to predict the time evolutions of the FCP and FTD of WDRWF flows. For FCP prediction, the performance of the Light Gradient-Boosting Machine (LightGBM) and multi-layer perceptron (MLP) is compared quantitatively. For the spatial-temporal evolution of FTD, a computationally efficient deep neural network architecture named ConvLSTM-AutoEncoder was developed. It employs ConvLSTM and convolutional encoder-decoder to reduce the dimensionality of high-dimension dynamic WDRWF flows and forecast the spatial-temporal characteristics aspects, which has significant potential in the application of determining when to switch on the active anti-/de-icing systems for better aircraft icing protection.

The proposed models are trained with experimental data,³⁸ and the test set data differ significantly from the training set data. The established strategies' generalizability and efficacy are assessed using error examination. The ConvLSTM-AutoEncoder model's robustness to noise is evaluated. While machine learning techniques have been used widely in recent years to study fluid dynamics problems, almost all the previous studies focused on single-phase fluid flows with numerical simulation results used to train and test ML models. To the

best of the authors' knowledge, the work reported here is the first of its kind to use ConvLSTM-AutoEncoder, MLP, and LightGBM trained by experimental datasets to predict the characteristics of complex multiphase flows such as WDRWF flows.

II. MACHINE LEARNING METHODS AND DATASETS

In this section, the experimental datasets of WDRWF flows used in training and testing the machine-learning models are briefly introduced first. Then, the background and hyperparameters of LighGBM and MLP models are given for FCP prediction. A new ConvLSTM-AutoEncoder architecture is proposed for FTD prediction of the WDRWF flows. The last two subsections introduce error evaluation techniques and the procedure for conducting FCP/FTD training and prediction.

A. Experimental datasets used to train and validate machine learning models

In the present study, the experiment datasets of Zhang *et al.*³⁸ on WDRWF flows were used to train and validate established machine learning models. While a brief description of the experiment study and the key features are provided below, further details of the experimental setup and measurement results can be found in Zhang *et al.*³⁸

Figure 1(a) depicts the experimental setup used to characterize the WDRWF flow. A flat plate was mounted on the bottom wall of a wind tunnel section. Wind-driven water runback flows over the test

plate are generated by pumping de-ionized (DI) water from a reserve tank through a row of tiny holes (i.e., the width of the tiny hole array $D = 60$ mm) drilled at the front of the test plate. The transient behavior of the wind-driven water film was measured quantitatively by using an advanced digital image projection (DIP) system.³⁸ The DIP system has a field of view of 100 mm in length and 60 mm in width. A digital camera is used for the DIP measurements at a frame rate of 30.0 Hz, with 600 snapshots being acquired for each test condition. The key parameters adjusted during the experiments include the incoming wind speed, V_∞ , and the water film flow rate, Q , over the plate surface. While the incoming airflow speed is changed from $V_\infty = 10$ to 20 m/s with an interval of 5 m/s, the water film flow rate is set to be 100 and 200 ml/min. Figure 1(b) depicts the representative results of the DIP measurements. The time evolution of the front contact line (FCL) of the wind-driven runback film flow can be located based on the time-resolved DIP measurements with a local water film thickness of 0.05 mm. In the present study, the front contact point (FCP) location refers to the utmost downstream position of the front contact line when multiple water rivulets are formed on the test plate. FCP is illustrated by point A' in Fig. 1(c).

Many variables influence the uncertainty level of DIP measurements, including the angle formed between the camera and the projector, the resolutions of the camera and projector used for the DIP measurements, the interrogation window size of the cross correlation process, the signal-to-noise ratio of the images, and the oscillation of

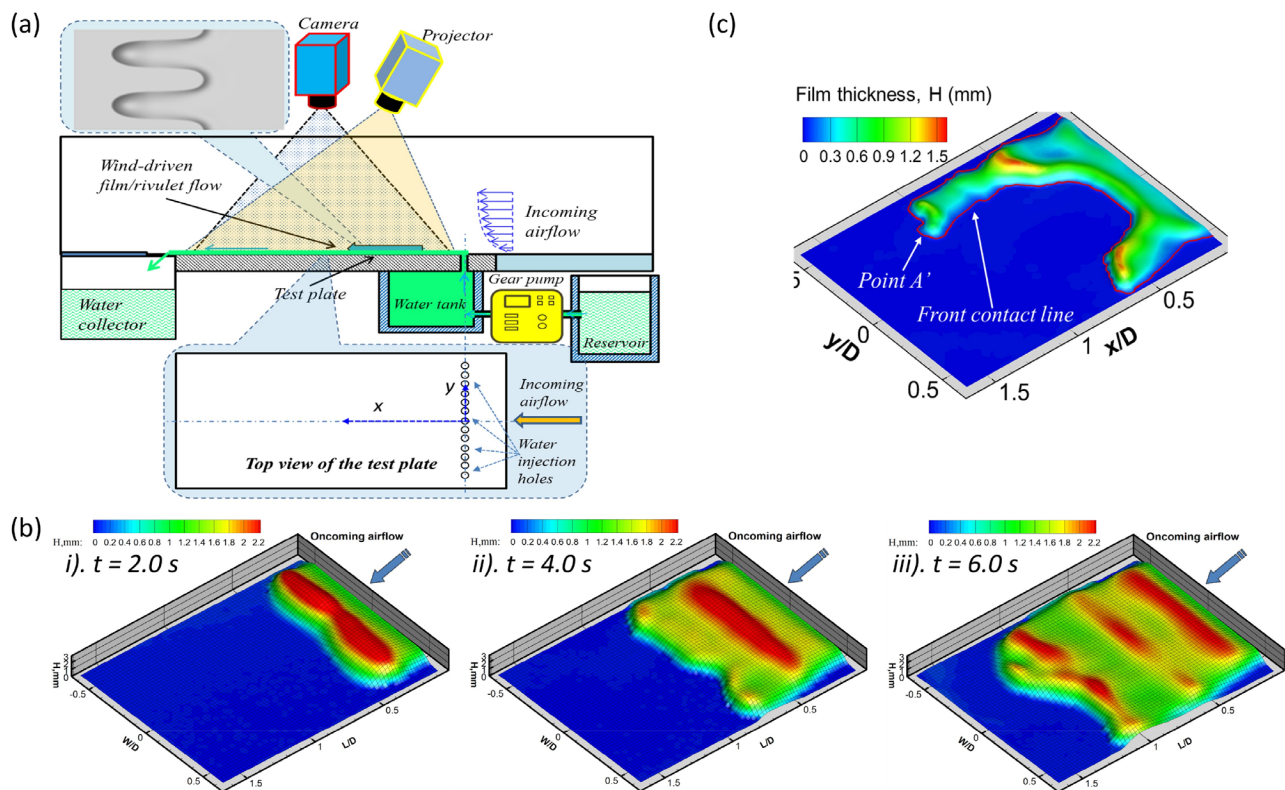


FIG. 1. DIP measurements of wind-driven water film/rivulet flows over a test plate:³⁸ (a) experimental setup; (b) time-resolved DIP measurement results for case #1 (as listed in Table I, $V_\infty = 10$ m/s and $Q = 100$ ml/min); and (c) schematic to define the front contact point (FCP).

TABLE I. A summary of the experimental dataset available for the data training of ML.

Index of case	Experimental conditions	Number of snapshots (FCP)	Number of snapshots (FTD)	Max thickness (mm)
1	$Q = 100 \text{ ml/min}, V_\infty = 10 \text{ m/s}$	200	200	2.71
2	$Q = 100 \text{ ml/min}, V_\infty = 15 \text{ m/s}$	42	60	1.63
3	$Q = 100 \text{ ml/min}, V_\infty = 20 \text{ m/s}$	34	70	1.02
4	$Q = 200 \text{ ml/min}, V_\infty = 10 \text{ m/s}$	59	60	2.72
5	$Q = 200 \text{ ml/min}, V_\infty = 15 \text{ m/s}$	27	50	1.51
6	$Q = 200 \text{ ml/min}, V_\infty = 20 \text{ m/s}$	16	40	0.85

the experiment platform.³⁸ The resolution restrictions of the camera and projector used for the DIP measurements would limit the spatial resolution of the measured water film thickness fields. For the DIP measurement results used in the present study, the smallest resolved grid of the measured water film thickness fields is $1 \times 1 \text{ mm}^2$.

Table I lists the six cases with distinct combinations of Q and V_∞ and the number of data files in each combination (or case). In the present study, fully developed WDRWF flow is the situation when the wetted area of the water film does not change with time in the field of view of the DIP measurements. Since the present study focuses on the under-development process of the WDRWF flow, instead of the fully developed WDRWF flow, the number of snapshots available for the ML model training is smaller than 600 for each case. The number of snapshots of FCP in each case is different because the time used by each case’s FCP to reach the end of the camera’s field of view is different, while the camera’s frame rate is fixed at 30 Hz. Less FCP snapshots mean less time for a water film to pass over the camera’s field of view. The reason why the number of data files of FTD is slightly more than that of FCP is that FCP datasets consist of only the developing phase of the water film flow, while the FTD datasets consist of both the developing phase and a part of the fully developed phase of the water film flow. The topology of the fully developed water film flow does not change significantly compared to the developing water film, and the shape of the contact line of the fully developed water film flow is almost unchanged compared to the developing water film. As a result, the fully developed phase of the water film flow is much easier to predict than the developing phase. If the proposed model can accurately predict the water film flow’s developing phase, so does the fully developed phase. That is why not all the fully developed phases of the water film flow are used to train and test the proposed model.

For FCP’s dataset, each snapshot is the location of the FCP. However, for the FTD dataset, each snapshot consists of the FTD over 5440 points in a two-dimensional region. The grid size of the 2D region is 68×80 , corresponding to a domain of $60 \times 80 \text{ mm}^2$ in the physical world. Before training, the datasets are split into a training set, validation set, and test set. Taking the setup 1 in Fig. 2 as an example to illustrate the details of dataset splitting method, to assess the model’s performance to predict the flow characteristics in case #1, snapshots in case #1 are set to be the test set, while the snapshots from test case #2 through test case #6 are set to be a “temporary” set. For each case contained in the temporary set, 80% of the snapshots were randomly sampled and then assigned to the training set (green block), while the remaining 20% were assigned to the validation set (blue block). The other five setups follow a similar splitting procedure, except that a different case is chosen as the test set, as shown in Fig. 2.

During training, for the six cases given in Table I, the characteristics of the wind-driven runback water film/rivulet flow vary greatly from case to case. To test the capacity and generalizability of the established model in forecasting the characteristics of WDRWF, the snapshots of five cases are used to train the model, while the trained model’s performance is assessed by predicting the remaining case.

B. The LightGBM model

LightGBM³⁹ is a machine-learning method for boosting gradient trees from end to end. It is intended for and frequently used by data science researchers to address supervised machine learning issues quickly and effectively. It uses a depth-constrained leaf-by-leaf development strategy, which means splitting one leaf node layer at a time, primarily by scanning each leaf individually to calculate the optimal gain and then repeating the process. This can simplify model

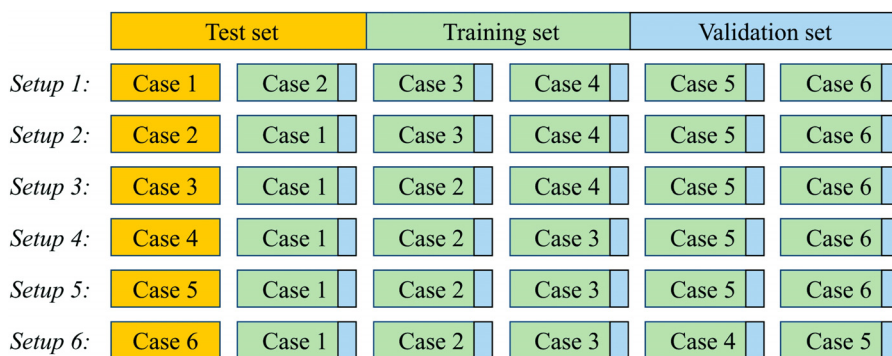


FIG. 2. The splitting method for the 6 wind-driven water film/rivulet flow cases listed in Table I.

complexity and reduce overfitting errors. Figure 3(a) depicts the method for growth. A detailed explanation of the computational techniques underlying LightGBM is beyond the scope of this study but can be found in cited source.²³ Using the boosting approach, a large number of decision trees are fitted to a training dataset. As shown in Fig. 3(a), the trees are built sequentially, with each successive tree aiming to reduce the residuals of the previous tree, allowing the model to learn more gradually. When adding trees yields no additional improvement in residuals, all trees are totaled to generate the final design. Gradient descent optimization is used to reduce model loss while adding more trees, giving rise to gradient boosting for the aforementioned boosting procedure.²⁰

It is challenging to optimize the LightGBM model’s hyperparameters, such as the number of trees, the depth of each tree, and the rate of learning. In the present study, the hyperparameter values were determined by a scikit-learn library called “GridSearchCV.”²⁵ To comprehend GridSearchCV, it is essential to introduce the concept of grid search first. Grid search evaluates all combinations of given hyperparameters and values to find the optimal combination. GridSearchCV combines grid search with cross-validation, commonly using K-fold cross-validation. This iterative approach divides data into N parts, using one for testing and N-1 for training in each cycle. The model’s performance is recorded and averaged across iterations. The combination with the best average performance will be selected. In this study, five-fold cross-validation and R^2 metrics served as the performance indicator are used. All datasets are fed into the GridSearchCV to find the optimal combination of hyperparameters. Noteworthy here is that although all datasets are used to determine the hyperparameter of the model, the trained model in GridSearchCV will not be saved nor used in the formal training and test stage. When formally training and evaluating the model, the data split procedure mentioned in Sec. II.A is conducted, as shown in Fig. 2. The following hyperparameters are determined by GridSearchCV. The learning rate is 0.004. The quantity of leaves is 128. The maximum depth and child weight of the decision trees are 6 and 4, respectively. The fraction of the feature is 0.8. The fraction of bagging is 0.6, with a frequency of once per 12 epochs.

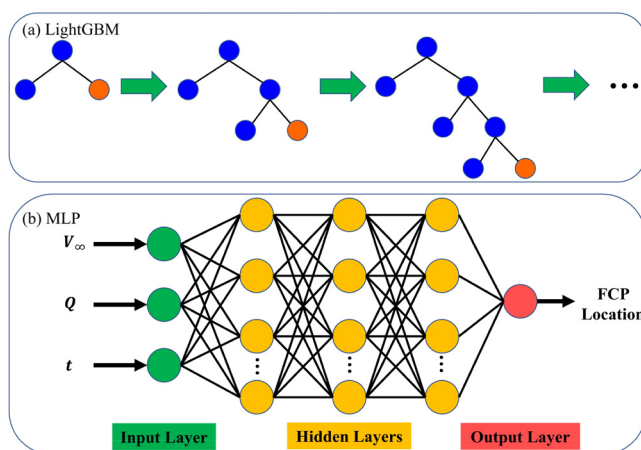


FIG. 3. Models used to predict FCP evolution. (a) LightGBM’s growth algorithms. Blue and orange indicate old and new nodes, respectively. (b) The MLP architecture.

The L_2 regularization coefficient is 0.001. Early stopping is used to adjust the number of epochs, which cannot be integrated into the GridSearchCV. So, the relevant hyperparameters are selected by lots of manual trials. The maximum number of epochs is set to be infinite, with the early stopping patience of 50 epochs. However, in practice, the training usually stops before 1000 epochs. The loss function is the mean absolute percentage error. On the scikit-learn platform,⁴⁰ the LightGBM algorithm was implemented. The input of the LightGBM model is the water flow rate, freestream velocity, and time. The output of the model is the corresponding location of FCP.

C. The MLP model

The MLP⁴¹ is a neural network made up of several neurons that are linked together to form a complex network. The neuron receives multiple input signals, which are then combined to form the output value. As a result, the neuron can be considered a cell that can map the nonlinear relationship between input and output. Figure 3(b) depicts the MLP design used in this study, which consists of multiple layers with several neurons. The MLP’s first layer receives input features, combines the inputs (freestream velocity, flow rate, and time), and passes the modified features through a nonlinear activation function. The modified output of the first layer is sent to the second layer, and the process is repeated layer by layer until the final layer predicts the output.³⁸ To optimize an MLP’s biases and weights, error backpropagation techniques are used.⁴² Literature⁴³ includes additional details about MLP. MLP model’s architecture (Table II) and a part of the hyperparameters are selected by using GridSearchCV. GridSearchCV determines the following hyperparameters. The learning rate is 0.001. The L_2 regularization coefficient is 0.001. Early stopping is used to adjust the number of epochs, which cannot be integrated into the GridSearchCV. So, the relevant hyperparameters are selected by lots of manual trials. The maximum number of epochs is set to be infinite, with the early stopping patience of 50 epochs. However, in practice, the training usually stops before 1000 epochs. The optimizer is Adam.⁴⁴ The loss function is the mean squared error. On the PyTorch Platform,⁴⁵ the MLP architecture was implemented.

D. The ConvLSTM-AutoEncoder architecture

For the FTD of unsteady WDRWF flow, a deep neural network architecture was developed to directly predict an FTD at a future time step using a sequence of FTDs in the past. The current state and historical motion determine the future film thickness distribution, as stated in the following equation:

$$P_{i+k} = f(P_i, P_{i-1}, P_{i-2}, \dots, P_{i-m}), \tag{1}$$

TABLE II. MLP network architecture.

Layer type	Number of neurons	Activation function
Input	3	Linear
Hidden-1	16	ReLU ³⁸
Hidden-2	64	Tanh ³⁸
Hidden-3	16	Sigmoid ³⁸
Output	1	Linear

03 October 2023 15:40:53

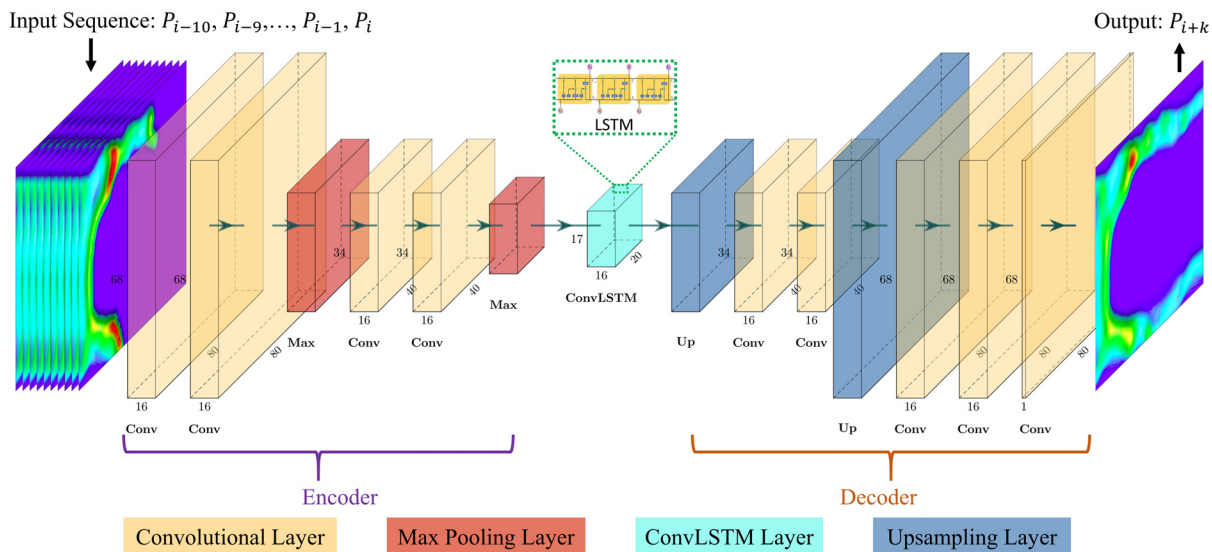


FIG. 4. Illustration of the ConvLSTM-AutoEncoder model used in the present study.

where P_i is the film thickness distribution at the time t_i , m denotes the number of prior FTDs used to estimate the FTD in the future, and k is the number of predicted future time steps by the model. For instance, $k = 5$ and $m = 10$ mean making a prediction of the FTD at t_{15} using film thickness distributions between t_0 and t_{10} if $i = 10$. A new architecture called ConvLSTM-AutoEncoder was established to perform the prediction task defined by Eq. (1), which is composed of the encoder, the ConvLSTM layer, and the decoder (see Fig. 4). To demonstrate the performance of the proposed architecture, $m = 10$ and $k = 1, 2, \dots, 10$ are used in the present study.

The encoder comprises two feature extraction blocks, as shown in Fig. 4 and Table III. Each feature block consists of two convolutional layers and a max pooling layer. It is fed with a sequence of high-resolution FTDs to generate dense, low-resolution key spatial features called latent features. The following ConvLSTM layer is fed with the encoder's output to predict new latent features at the future time step. The predicted future latent features are then fed into the following decoder, which comprises two decoding blocks, each with two convolutional layers and an upsampling layer. The decoder can recover high-dimensional FTDs from dense low-dimensional latent features. The combination of encoder and decoder is called an AutoEncoder.⁴² The architecture has 35,041 trainable parameters in total for the datasets of the present study.

The architecture and some of the ConvLSTM-AutoEncoder model's hyperparameters are selected by using GridSearchCV. However, the optimal combination may not be selected if the hyperparameter makes the model highly time-consuming. The batch size is set to 20. The L_2 regularization coefficient is 0.001, which is applied to the kernels of the convolutional layers to overcome overfitting. A sequential learning rate decaying technique is used to adjust the learning rate, which cannot be integrated into the GridSearchCV. So, the relevant hyperparameters are selected by lots of manual trials. The following hyperparameters are chosen from manual trials. The initial learning rate is 0.001. The decay rate of the learning rate is 0.9, with a

decay patience of 50 epochs. The early stopping patience is 200 epochs, which helps prevent overfitting. The Adam optimizer is employed during the training procedure.⁴⁴ The loss function is the mean absolute error. On the TensorFlow platform,⁴⁶ the ConvLSTM-AutoEncoder architecture was implemented.

The components involved in the ConvLSTM-AutoEncoder architecture are briefly introduced here. LSTM is effective at capturing temporal characteristics.²⁸ The forget gate, output gate, and input gate are the three gates found in a typical LSTM cell. The LSTM cell regulates an input data stream by adding, removing, or passing it on to the next cell. As shown in Fig. 5, LSTM uses its internal storage to generate estimations based on the most recent input sequence context rather than the most recent input. The original LSTM cell's hidden and input

TABLE III. ConvLSTM-AutoEncoder architecture parameters.

Layer name	Kernel dimension/stride	Output dimension
Conv1	$3 \times 3 / 1$	$68 \times 80 \times 16$
Conv2	$3 \times 3 / 1$	$68 \times 80 \times 16$
MaxPooling1	$2 \times 2 / 2$	$34 \times 40 \times 16$
Conv3	$3 \times 3 / 1$	$34 \times 40 \times 16$
Conv4	$3 \times 3 / 1$	$34 \times 40 \times 16$
MaxPooling2	$2 \times 2 / 2$	$17 \times 20 \times 16$
ConvLSTM	$3 \times 3 / 1$	$17 \times 20 \times 16$
UpSampling1	$2 \times 2 / 2$	$34 \times 40 \times 16$
Conv5	$3 \times 3 / 1$	$34 \times 40 \times 16$
Conv6	$3 \times 3 / 1$	$34 \times 40 \times 16$
UpSampling2	$2 \times 2 / 2$	$68 \times 80 \times 16$
Conv7	$3 \times 3 / 1$	$68 \times 80 \times 16$
Conv8	$3 \times 3 / 1$	$68 \times 80 \times 16$
Conv9	$3 \times 3 / 1$	$68 \times 80 \times 1$

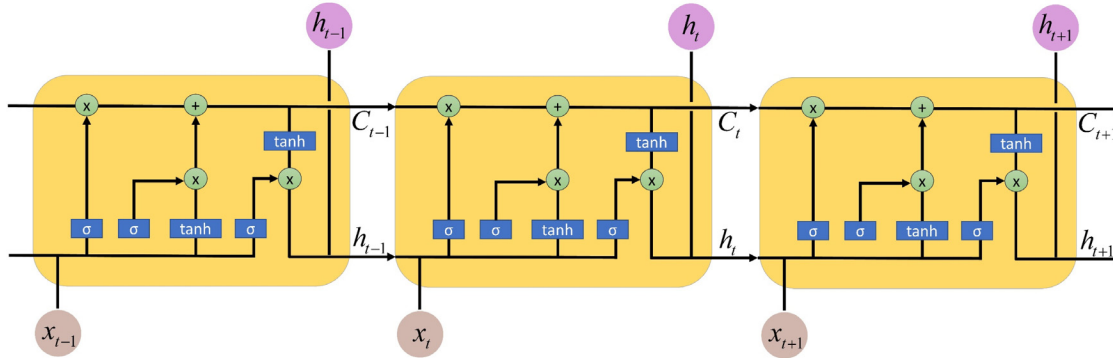


FIG. 5. The configuration of the LSTM cells.

states are both 1D arrays, which makes it impossible to characterize the underlying spatial structures of 2D data, lowering estimation accuracy. Shi *et al.*⁴⁷ presented the ConvLSTM module, which can process 2D input states while preserving the spatial structures of 2D data by modifying the gates to have 2D data dimensions. As a result, a 2D image could be fed into the ConvLSTM module to generate a 2D cell state.

As shown in Eq. (2), the ConvLSTM's input gate is represented by ϑ , the output gate is symbolized by the symbol ψ , and the forgotten gate's symbol is F . The state of the cell is denoted by C , the output of the cell is indicated by h , and the input of the cell is marked by x . Each gate's weight is represented by W . The Hadamard product is symbolized by \odot , and the symbol for the convolution manipulation is $*$,

$$\begin{cases} \vartheta_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \odot C_{t-1} + b_i), \\ F_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \odot C_{t-1} + b_f), \\ C_t = F_t \odot C_{t-1} + \vartheta_t \odot \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c), \\ \psi_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \odot C_t + b_o), \\ h_t = \psi_t \odot \tanh(C_t). \end{cases} \quad (2)$$

Now the convolutional layer is introduced. The convolution layer is responsible for extracting image characteristics and improving the signal, while reducing the background noise.⁴⁸ The parameters of the convolutional layer are made up of a series of learnable kernels with a narrow receptive field and the same depth as the input volume. Each kernel is convolved across the width and height of the input volume during the forward pass, computing the dot product between the kernel's entries and the input and generating a two-dimensional activation map. The entire output volume of the convolution layer is created by stacking the activation fields for each kernel along the depth dimension. A convolutional filtering operator is used by the convolutional layer. The previous layer's output feature field is generated using the following nonlinear activation function:

$$\hat{y}_c = \sigma(K * x_c + b_c), \quad (3)$$

where σ is a function of nonlinear activation, $*$ represents the operation of convolution, \hat{y}_c represents the feature field of the output, x_c are input features for the convolutional layer, K is the kernel for convolutional learning, and b_c reflects the bias. The ReLU function is used as

the nonlinear activation function in this study for two reasons. First, the ReLU function's output is positive, which corresponds to the range of FTD values. Second, the ReLU function's performance was compared to that of two other advanced ReLU-like activation functions, PReLU⁴⁹ and SoftPlus,⁵⁰ and discovered that the ReLU function has the highest accuracy. Section III B 2 describes the comparison results in detail.

The max pooling layer⁴⁶ method is also used in the present study. Pooling may effectively reduce the model's complexity and calculations to prevent overfitting and maintain the invariance of crucial input feature maps.⁴⁷ Max pooling down-samples the original input by applying a max filter to non-overlapping subregions. It can restore texture features, which is essential for capturing the position of the contact line of the WDRWF flow. A further advantage is its ability to denoise noisy data, which is crucial for this study because the ConvLSTM-AutoEncoder architecture will be trained and evaluated using experimental data.

E. Error evaluation techniques

Three error metrics (R^2 , relative error, and absolute error) are used to evaluate the effectiveness of the established machine learning models in forecasting the development of the FCP and FTD.

1. R^2 metric

The R^2 metric aims to quantify the degree to which the forecasts and the actual data on the ground were related to one another, which is widely used in the machine learning community. When the value of R^2 is closer to 1, the model's performance is better. R^2 is used to assess the FCP prediction performance and is exemplified by the following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{f}_p(V_\infty, Q, t_i))^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (4)$$

where y_i is the true FCP position at a certain time moment, $\hat{f}_p(V_\infty, Q, t_i)$ is the model's prediction for FCP position at t_i , N is the number of samples in each case, and \bar{y} is the average of N true FCP

positions. If R^2 is greater than 0.85, then there is a good agreement between the two groups.²⁵

2. Relative error metric

The relative error, which aims to quantify how large the error is relative to the true value, is widely used to characterize the uncertainty level of the measurements. It is computed by dividing the difference by the ground-truth data and then calculating the absolute magnitude of the result. The following formula is usually used to characterize the relative error magnitude at t_i when the freestream velocity is V_∞ and the flow rate is Q :

$$\epsilon_{rel,i} = \left| \frac{y_i - \hat{f}_p(V_\infty, Q, t_i)}{y_i} \right|, \quad (5)$$

where y_i is the true FCP position at t_i , $\hat{f}_p(V_\infty, Q, t_i)$ is the model's prediction for FCP position at t_i , $i \in [1, N]$, and N is the number of samples in each case. Because the error distributions of FCP prediction were highly skewed, the median relative error (MRE) rather than the mean relative error was used to evaluate the efficacy of the models in forecasting the development of FCP.

3. Absolute error metric

For FTD's error evaluation, the absolute errors were used as metrics because the uncertainty level of the experimental data is known, which is about 0.02 mm. Absolute error attempts to quantify the difference between the actual and predicted water film thickness at a certain location at a certain time moment. An absolute error map at a certain time moment could be obtained by displaying the absolute error at every point in the domain, as shown in the third column of Fig. 9. The following formula is used to characterize the absolute error magnitude at a certain location at a certain time moment:

$$\epsilon_{i,j}^{abs} = |h_{i,j} - \hat{f}_h(x_{i,j}, y_{i,j})|, \quad (6)$$

where $h_{i,j}$ is the true thickness at the location $(x_{i,j}, y_{i,j})$ at a certain time moment, and $\hat{f}_h(x_{i,j}, y_{i,j})$ is the model's prediction for water film thickness at the location $(x_{i,j}, y_{i,j})$ at a certain time moment.

F. The procedure of training and prediction

1. FCP prediction

Table IV depicts the method used in the present study for developing machine learning-based models for forecasting the development of the FCP position. The established models required time, freestream wind velocity, and water flow rate as inputs. The models' outputs were the corresponding positions of FCP. Before training, the datasets were normalized by dividing the FCP positions by the dimension of the camera's field of view and dividing the time by the time it takes for the water film to flow across the entire field of view of the camera.

For the test cases of the present study, the training time at 2.30 GHz Intel Xeon Gold 6140 CPU is approximately 1.5 s for LightGBM and 1.7 s for MLP. The wall time of predicting the FCP evolution of a single case is approximately 20 ms for LightGBM and 6 ms for MLP.

TABLE IV. Procedure for constructing ML models to make FCP predictions.

Procedure of Front Contact Point (FCP) Prediction	
Data:	Time-resolved FTD measured by the DIP system.
Results:	FCP location predictions under given conditions of incoming air speed, flow rate, and elapsed time.
Step 1:	Extract the FCP location from the time-resolved thickness distribution to form the FCP dataset.
Step 2:	Assign the corresponding freestream velocity and flow rate to each sample in the FCP dataset.
Step 3:	Split the FCP dataset into 6 setups consisting of training, validation, and test sets in each setup. (See Fig. 2.)
Step 4:	Normalize the exposure time and FCP locations.
Step 5:	Construct an XGBoost/LightGBM/MLP model for each of the 6 setups
Step 6:	Perform cross-validation experiments to obtain the optimal hyperparameter for the model.
Step 7:	Estimate the location of FCP by feeding the velocity, flow rate and elapsed time into the trained model.
Step 8:	Assess the model's accuracy by the established error evaluation methods.

2. FTD prediction

Table V illustrates the procedure for constructing ML models to predict FTD. The ConvLSTM-AutoEncoder maps a sequence of past FTDs to a future sequence of FTDs. Therefore, the dataset must be first transformed into multiple short sequences. In each case, the original sequence of FTDs was rearranged into multiple short sequences of input/output pairs. In this study, 10 time step FTDs were selected as an input sequence and 10 shifted FTDs were used as the model's output sequence. The time difference between the first FTD of the input sequence and that of the shifted output sequence is defined by k in Eq. (1). The duration of each time step in the dataset is 1/30 s. Then, only the final FTD in the output sequence will be used to evaluate the model's performance. In short, the ConvLSTM-AutoEncoder receives 10 sequential FTDs from previous periods as input, and the output is the FTD at $t = i + k$. It operates in a manner like the weather forecast. The estimation's accuracy could be determined by comparing the estimated sequence to the true sequence.

On a 2.30 GHz Intel Xeon Gold 6140 CPU, the training time for each model ranges from 6 to 12 h. The wall time for predicting a single FTD is approximately 80 ms. The model's input data are normalized using Min-Max normalization.⁵¹ Before evaluating the model's performance, the output will be restored to its original scale.

To demonstrate the convergence of the proposed models and the effectiveness of the early stopping regularization, representative training histories of setup 4's LightGBM model, MLP model, and ConvLSTM-AutoEncoder model ($k = 1$) are displayed in Fig. 6. The convergence curves of other setups were found to have a similar trend, so only the curves of this setup were illustrated here for brevity. The curves show that as the loss converges and has no more improvement, the early stopping regularization stops the training.

TABLE V. Procedure for constructing ML models to make FTD predictions.

The procedure of FTD prediction

Data: Time-resolved FTDs measured by the DIP system.

Results: Predictions of the FTDs at $t = i + k$ corresponding to the given sequence of FTDs at $t = i - 10, i - 9, \dots, i$.

Step 1: Transform the dataset listed in Table I into multiple pairs of input/output short sequences with a length of $m = 10$ time steps for each sequence of the pair.

Step 2: Split the transformed pair dataset into 6 setups consisting of training, validation, and test sets in each setup. (See Fig. 2.)

Step 3: Normalize the thickness by Min-Max normalization.

Step 4: Construct a ConvLSTM-AutoEncoder model for each of the 6 setups

Step 5: Perform cross-validation experiments to obtain the optimal hyperparameter for the model.

Step 6: Estimate the FTD at $t = i + k$ by feeding a sequence of thickness distributions at $t = i - 10, i - 9, \dots, i$ into the trained model.

Step 7: Concatenate the model outputs in the order of time to compose an estimated sequence of water film thickness distributions.

Step 8: Assess the performance of the model by established error methods.

III. RESULTS AND DISCUSSION

For FCP prediction, the prediction performance of LightGBM and multi-layer perceptron (MLP) were compared. Both models were found to perform well in capturing intermittent and smooth features in general. For the spatial-temporal evolution of thickness distribution, the performance of ConvLSTM-AutoEncoder was tested by varying the predicted time steps and adding noises to the input sequence. The intermediate features of the ConvLSTM-AutoEncoder were also analyzed to interpret the mechanism of this established architecture.

A. Forecasting of front contact point (FCP) evolution

Figure 7 shows the values of R^2 and MRE for the LightGBM and MLP models for the six tested cases listed in Fig. 2. The mean values of R^2 and MRE of the tested cases are listed in the legend. In terms of overall performance, the LightGBM model was found to outperform the MLP, with average values of $R^2 = 0.96$ and $MRE = 0.090$.

For brevity, the predicted and true development of FCP location in only two representative setups 4 and 6 are plotted to compare the performance of the two models (see Fig. 8). The black and red lines illustrate the evolution of the FCP trajectory measured in the experiments and predicted by models, respectively. To assess the accuracy of the predictions, blue relative error bands are generated at each data point. As shown clearly in Fig. 8, the predicted evolution of FCP and

the experimental results were consistent in general. Only the LightGBM models captured the “deceleration–stagnation–acceleration” behavior,³⁸ which refers to the film heads running back in a stumbling motion, as shown in Figs. 8(a) and 8(b). This is because LightGBM consists of tons of discrete decision trees, which can fit non-smooth data well. In contrast, the tanh and sigmoid activation functions utilized by the MLP model are excessively smooth; consequently, the estimations of the MLP model in Fig. 8 lack deceleration–stagnation–acceleration cycles. The force balance analysis³⁸ can explain the underlying physical mechanism of the deceleration–stagnation–acceleration cycles shown in Fig. 8. The water film obeys the basic law that it moves forward when the aerodynamic forces (primarily the drag force) exceed the restraining surface tension force. The film head is found to be lower in height than the film/rivulet body just before it stagnates. The corresponding aerodynamic drag force on the thin film head is less than the surface tension force. As a result, the film head is decelerated until it becomes stationary. The surface wave from upstream arrives at the film head immediately after it has stagnated. It raises the height of the film head, which significantly raises the aerodynamic drag. The increased aerodynamic drag easily overcomes the surface tension force, accelerating the film head. Because of the rapidly increased wetted area swept by the accelerated film head, the film head’s height decreases before the next surface wave arrives. The corresponding aerodynamic force decreased and

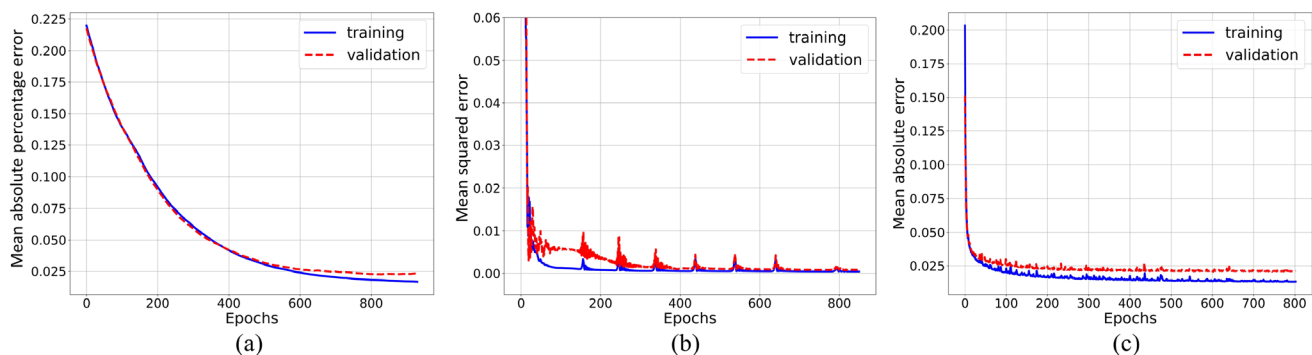


FIG. 6. Representative learning curves of the (a) LightGBM, (b) MLP, and (c) ConvLSTM-AutoEncoder models.

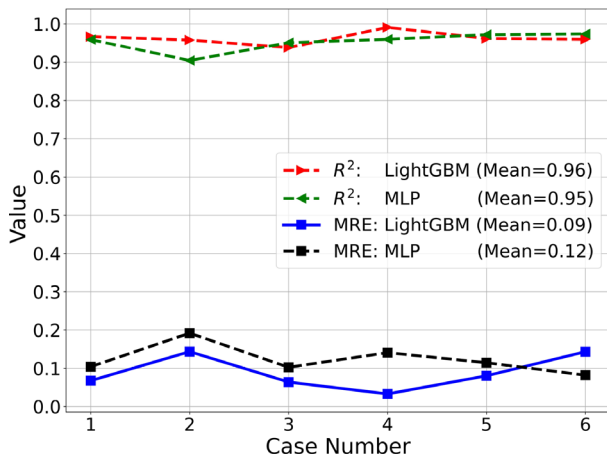


FIG. 7. A comparison of the performance of LightGBM and MLP in predicting FCP. The mean value in the legend is the average value of the 6 tested cases.

became smaller than the restraining surface tension, causing the film head to decelerate. The preceding process then iterates to create the deceleration–stagnation–acceleration cycles.

Based on the solid blue line in Fig. 7, the LightGBM was found to be the best in predicting case #4. This is because, as shown in Fig. 8(a), the true profile of case #4’s FCP evolution history has the most “stagnation” steps of any other cases (not shown here), extending the

decision-tree-based method’s superiority in estimating discrete functions. In contrast, MLP performs best in predicting case #6, based on the black dotted line in Fig. 7. It is because the profile of the FCP evolution history of case #6, as shown in Fig. 8(d), is the “smoothest” among all cases, thereby enhancing MLP’s ability to predict continuous functions. The LightGBM predicts case #6 slightly less accurately than the MLP. Case #6’s true FCP trajectories are significantly more nonlinear than those of case #4. After the normalized time of 0.8, the slope of case #6’s trajectory is significantly steeper than before the normalized time of 0.6. The MLP’s prediction captures this slope difference. In contrast, LightGBM’s prediction throughout the time range has a nearly uniform slope equal to that of true FCP trajectories after 0.8 normalized time. This is because while LightGBM can capture nonlinear relationships, MLP surpasses it in superiority because of its utilization of multiple hidden layers, which enables the learning of hierarchical representations of data containing intricate patterns. Consequently, each model has its own advantages under specific conditions.

Observing the distribution of the relative error bands shown in Fig. 8 reveals that most of the large relative inaccuracy is due to small position values in the initial evolution stage. A minor bias of the predicted position could result in a substantial relative error for the extremely small values; for instance, if the actual location was 0.001 mm, the relative error of a predicted position of 0.002 mm could be 100%. However, an absolute error of 0.001 mm is quite small. That is why the model’s performance is evaluated by median relative error and R^2 scores, as shown in Fig. 7. In summary, the above error

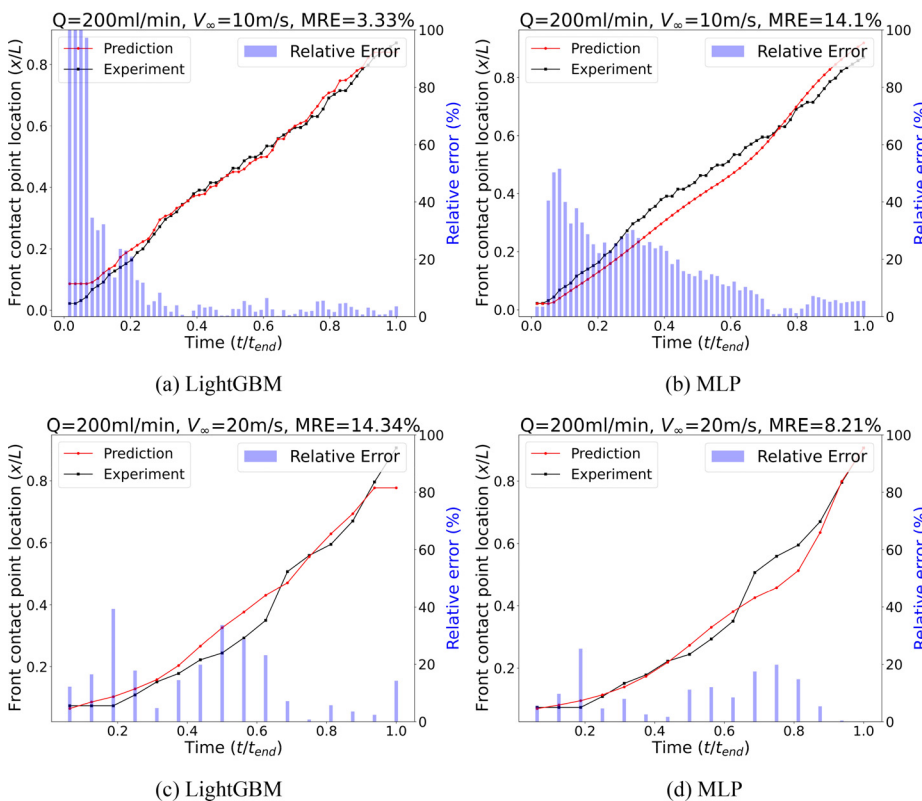


FIG. 8. Comparison of the FCP trajectories for case #4 ($Q=200\text{ ml/min}$, $V_\infty=10\text{ m/s}$) and case #6 ($Q=200\text{ ml/min}$ and $V_\infty=20\text{ m/s}$) predicted by LightGBM and MLP.

assessments reveal that the LightGBM and MLP performed well in predicting the evolution of the FCP of the WDRWF flow.

B. Forecasting the evolution of film thickness distribution (FTD)

Since the FCP evolution represents only the local dynamics of the fast-moving portion of the WDRWF flow, it is necessary to develop a machine learning-based model capable of capturing the global features of the WDRWF flow. The ConvLSTM-AutoEncoder architecture introduced in Sec. II D was used in the present study to estimate the evolution of the FTD in this section.

1. Assessment of the ConvLSTM-AutoEncoder's accuracy in predicting the FTD characteristics

In this subsection, the results from the model with $k=1$ are used to demonstrate the performance of the proposed model. The results and evaluation of the models with larger k are presented in the next subsection. Absolute-based error metrics are used to evaluate the performance of the proposed ConvLSTM-AutoEncoder model because (1) the disadvantage of relative error in evaluating small absolute errors discussed in Sec. III A; (2) the water film flows in the experimental dataset are essentially thin films with a small magnitude of thickness; and (3) the uncertainty level of experimental data is known. The difference between the true and predicted FTDs is collected over the entire predicted period to calculate the error metrics in Table VI. The training and validation set of the trained model does not contain the data from the tested case. Given that the DIP's uncertainty level is around 0.02 mm,³⁸ the median absolute errors (MAE) and average absolute errors (AAE) in Table VI are on the same order of magnitude as the uncertainty level. It indicates that the established ConvLSTM-AutoEncoder architecture is capable of accurately forecasting the thickness distribution evolution of an unseen WDRWF flow.

The accuracy of the ConvLSTM-AutoEncoder model is also demonstrated in Fig. 9 by displaying typical results of water film thickness distribution estimation for setups #1, #2, and #5 (defined in Fig. 2). Beginning on the left of Fig. 9, the true FTD from experiments, prediction outputs from the ConvLSTM-AutoEncoder model, and the absolute error maps were displayed. As expected, all forecast findings captured wind-generated surface waves very well. The high absolute error regions are generally located in the thick film region. This is because the aerodynamic shear force on the higher water film surface is greater, generating irregular surface waves at a high velocity. The complex interaction between the irregular surface wave and the airflow

makes it more difficult to predict the topology of the water film surface. By comparing Figs. 9(a) and 9(b), it can be concluded that increasing the freestream velocity leads to the transition from film flow to rivulet flow. The fundamental physical mechanism of this transition entails a complex interplay between aerodynamic drag force, surface tension, and dynamics of the free surface. The aerodynamic drag force is composed of shear force and pressure force. The aerodynamic shear force arises on the surface of the water film due to the velocity disparity between the wind and the water film. The application of shear force induces wave propagation on the surface of the water, while the cohesive forces of surface tension counteract the deformation of the water film. The velocity of surface waves exceeds that of the film body due to the exerted shear forces. As the velocity of the wind rises, a critical threshold is attained wherein the surface tension becomes inadequate to sustain the uniform structure of the film front, particularly when a high surface wave encounters the film front accompanied by significant aerodynamic drag forces. At this particular threshold, the portion of the film front with the highest shear force tends to be the first and most significant to be disrupted. These localized disruptions form the initial points of rivulet formation. Surface tension induces a preferential water flow toward the rivulets because rivulets exhibit lower energy than film flow.

The effect of flow rate is clearly shown in Figs. 9(b) and 9(c), where both the width and the number of rivulets were found to increase with the increasing water flow rate. The interaction between aerodynamic drag and surface tension can explain the mechanism underlying the increase in width. The wind exerts a drag force on the water's surface film, endeavoring to move it along. In contrast, surface tension resists this motion by attempting to maintain the water film's integrity and minimize its surface area. At low flow rates, surface tension prevails, forming rivulets with a narrower width. As the flow rate increases, the height of the film/rivulets increases, resulting in a more significant drag force, which can surmount surface tension, resulting in wider rivulets. The mechanism of an increase in the number of rivulets can be elucidated by the amalgamation of the instability induced by the shear force and the transition mechanism discussed in the preceding paragraph. The shear force induces disturbances in the water film. An increased water flow rate results in an augmented water film thickness. An increase in the thickness of the water film results in a corresponding amplification of the shear force, thereby causing the generation of more irregular surface waves with higher amplitude. When the irregular and high surface waves encounter the film front, a greater number of points on the front surpass the disruption threshold and subsequently generate rivulets compared to a film with a smaller thickness but moving at the same speed. Again, since the trained ConvLSTM-Autoencoder models were never fed with the data of unseen tested cases, the results in Fig. 9 show the model's remarkable generality in predicting a WDRWF flow whose features are very different from that of the training set.

2. The effect of predicted time steps k on the FTD prediction

According to Eq. (1) and Fig. 4, k represents the difference between the time step index of the last FTD in the model's input sequence and the time step index of the model's output. To investigate the predictive capability of the ConvLSTM-AutoEncoder model, 180

TABLE VI. Evaluation of the model's performance in estimating the unseen FTDs.

Case number	MAE (mm)	AAE (mm)
1	0.006	0.056
2	0.004	0.033
3	0.011	0.028
4	0.005	0.057
5	0.011	0.050
6	0.010	0.031
Average	0.008	0.043

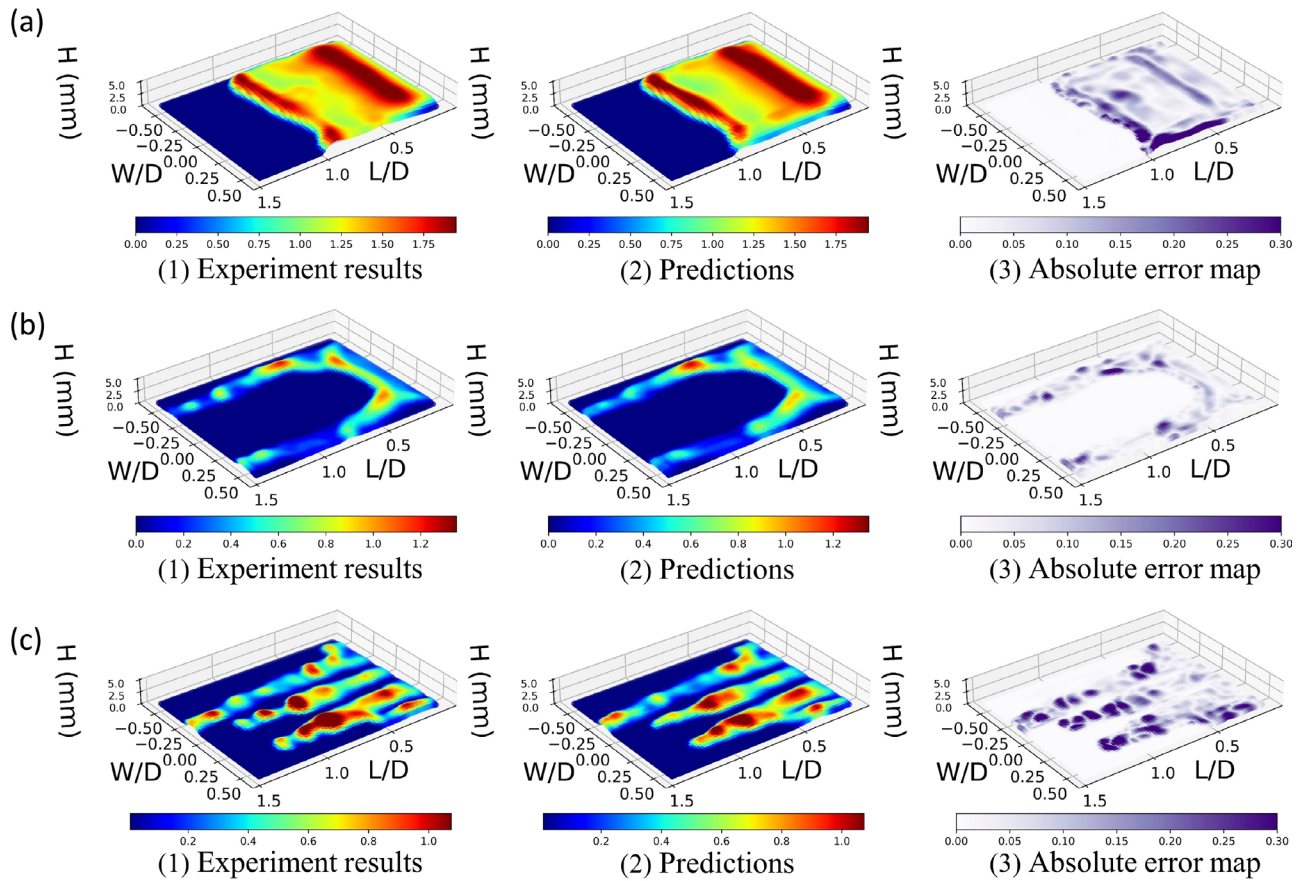


FIG. 9. Evaluation of the predicted film thickness for (a) case #1 ($Q = 100$ ml/min and $V_\infty = 10$ m/s) at $t = 4.60$ s; (b) case #2 ($Q = 100$ ml/min and $V_\infty = 15$ m/s) at $t = 1.80$ s; and (c) case #5 ($Q = 200$ ml/min and $V_\infty = 15$ m/s) at $t = 1.07$ s.

training and predicting experiments were conducted for all six setups (see Fig. 2) with ten distinct numbers of predicted time steps ($k = 1, 2, 3, \dots, 10$) and three distinct activation functions (PReLU, ReLU, and SoftPlus). Figure 10(a) depicts the effect of the number of predicted steps on AMAE, which is the case-average MAE among the six tested cases. Notably, all AMAEs of the ReLU function are within 0.02 mm, which is the experimental data’s uncertainty level. ReLU’s results are the smallest at every k , compared with the other two activation functions. This validates the selection of ReLU in this study, as discussed in Sec. II D. Moreover, for each individual line in Fig. 10(a), the AMAE increases as the number of predicted time steps increases. This trend is because the uncertainties in WDRWF flow increase with the increase in the predicted time step. First, the complexity and turbulence level of the boundary layer airflow are increased by interacting with the surface wave of the water film. This increased complexity of airflow increases the uncertainty in the development of the wind-driven water film in turn. Second, as the number of predicted time steps k increases, the possible types of the development of the wind-driven water film increase dramatically, widening the gap between the current film pattern and the predicted future film pattern. The dataset’s sampling frequency used in the present study is 30 Hz (i.e., the time step equals 0.033 s). With $k = 10$, the model can predict the FTD in the next 0.33 s.

Based on the experimental investigation of the dynamic icing process over the surface of a NACA0012 airfoil model,⁵² the ice accretion caused by runback water rivulet can move about 20% airfoil chord length (i.e., the airfoil chord length being 152.4 mm) toward the trailing edge in 0.33 s under a glaze ice condition of $LWC = 4.0$ g/m³, $T_\infty = -8.0$ °C, and $V_\infty = 50$ m/s. As a result, the current model has the potential to predict large changes in ice accretion and enable the icing protection system to be activated in advance to mitigate ice formation and accretion on airfoil/wing surfaces.

The contour of the distribution of MAE is presented in Fig. 10(b) to expand the case dimension of the ReLU curve in Fig. 10(a). While the horizontal axis represents the tested case, the vertical axis represents the number of predicted time steps k , and the value of MAE colors the map. Similar to Fig. 10(a), each case’s MAE increases with the number of predicted steps. Interestingly, cases #2 and #4 have a low error at every k , indicating that the models tested on these cases have uniform accuracy regardless of k . Our explanation for this phenomenon is that the flow features of the two cases may also be contained in their corresponding training set. For the other four cases, when $k = 5$, the MAEs were found to be less than the measurement uncertainty of the experimental data (i.e., ≤ 0.02 mm). When $k = 10$, the MAEs have the same order of magnitude as the measurement uncertainty.

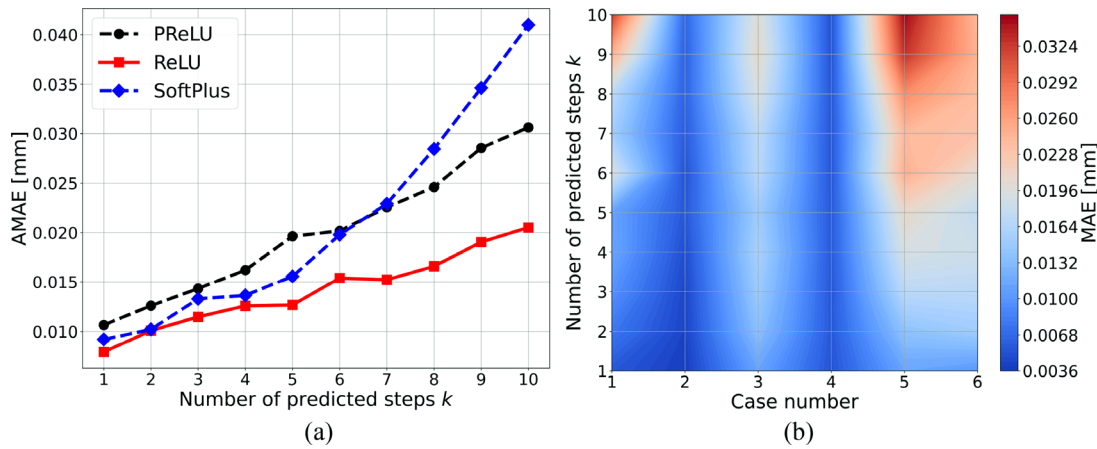


FIG. 10. The dependence of the prediction error on the number of predicted steps. (a) Comparison of AMAE of three different activation functions with respect to the number of predicted steps. (b) The contour map of the MAE distribution with respect to the number of predicted steps and the case number.

These results indicate that the proposed ConvLSTM-AutoEncoder model has a high degree of generalizability.

In order to evaluate the model’s performance on temporal characteristics prediction, a point (i.e., at $W/D = -0.37$ and $L/D = 0.40$) in the flow field was arbitrarily selected to reveal evolution characteristics of the water film thickness at that point for case #1 (i.e., $Q = 100$ ml/min and $V_\infty = 10$ m/s) with $k = 1, 5,$ and 10 in Figs. 11(a)–11(c).

The results of other points and other cases were found to have very similar features as this sample point, so only the results at this arbitrarily selected point were illustrated here for brevity. As shown in Figs. 11(a)–11(c), one of the fronts of the water film/rivulet flow reached the point of $W/D = -0.37$ and $L/D = 0.40$ around $t = 1.6$ s, after which the water film thickness was found to increase rapidly from 0 to around 1.8 mm and then oscillated around 1.8 mm.

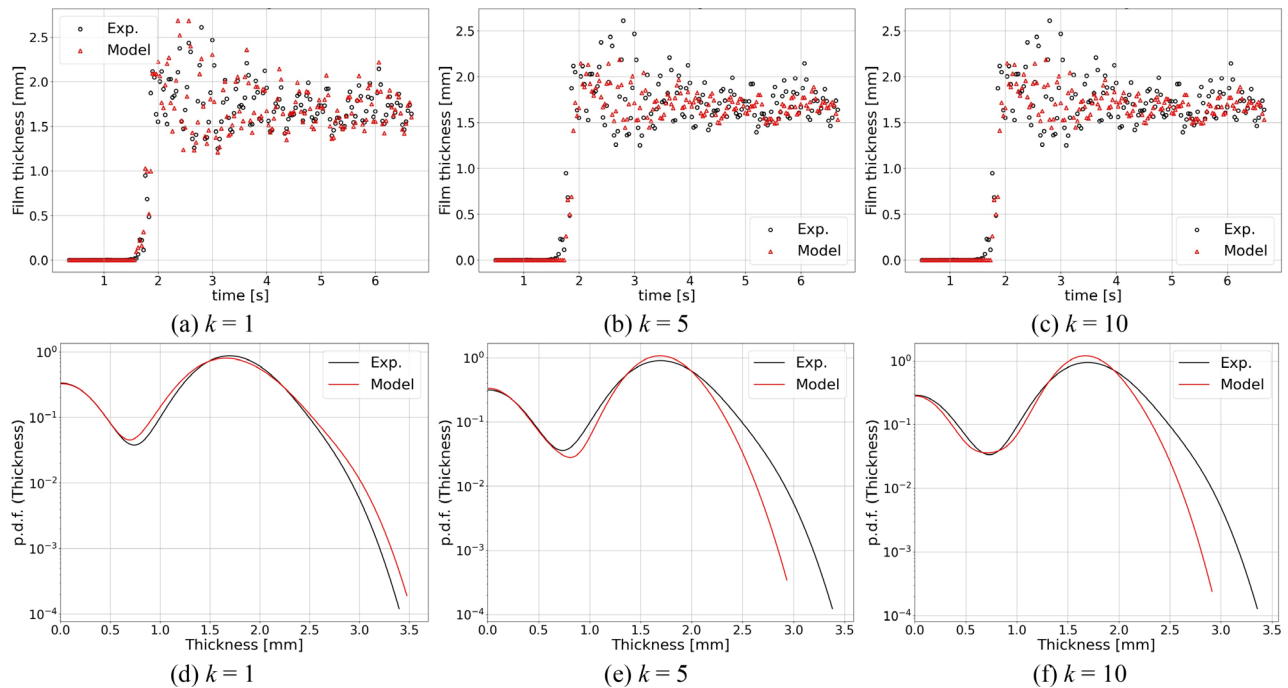


FIG. 11. Comparing the time sequences of the thickness between the experimental data and model prediction at a sample point of $W/D = -0.37$ and $L/D = 0.40$ for the tested case #1 ($Q = 100$ ml/min and $V_\infty = 10$ m/s) with different numbers of predicted time steps k . (a)–(c) are film thickness time series of the sample point when the number of predicted time steps is $k = 1, k = 5,$ and $k = 10$, respectively. (d)–(f) are the probability density function (PDF) of the film thickness time series of the sample point when the number of predicted time steps is $k = 1, k = 5,$ and $k = 10$, respectively.

This trend was captured by all the predictions with $k = 1, 5,$ and $10,$ but with varying degrees of accuracy. To assess the accuracy of the predictions more clearly, the probability density functions (P.D.F) of the time series at the sample point were calculated, which are shown in Figs. 11(d)–11(f). When the film thickness is between 0 and 2.0 mm, the P.D.F of all three models ($k = 1, 5,$ and 10) agrees with the experimental data well. For thicknesses greater than 2.0 mm, the discrepancy between the P.D.F of the model and that of the experimental data grows as the thickness increases. Because the highest thickness in case #1 is greater than that of other cases in the training dataset, this disparity grows as k increases, implying that the model’s accuracy degrades primarily due to the error in predicting the bigger film thickness.

To increase the interpretability of the proposed ConvLSTM-AutoEncoder model, the intermediate feature maps of the second, fourth, sixth, and eighth convolutional layers are given in Fig. 12 to illustrate how the neural network performs prediction tasks. The second and fourth convolutional layers are before the ConvLSTM layer, while the sixth and eighth convolutional layers are after the ConvLSTM layer. The feature maps depicted in Fig. 12 were chosen because the average intensity is the highest among the feature maps in their respective convolutional layers. Inputs to the model are a sequence of film thickness distributions from $t = 0.367$ to $t = 0.66$ s; however, for simplicity, only the last FTD ($t = 0.66$ s) of the input series is shown in Fig. 12. The model’s output is at $t = 0.7$ s for $k = 1.$ The model’s output is at $t = 0.83$ s for $k = 5.$ The intensity of the feature maps is positive because the activation function used in this study is the rectified linear unit (ReLU). So, the greater the intensity, the greater the interest. The feature maps of the second and fourth convolutional layers show more interest in the shape of the water film in the last FTD of the input time series. The fourth convolutional layer’s resolution is less than that of the second convolutional layer because the dimensions in the encoder structure are reduced to focus on critical features. The ConvLSTM is trained to learn the spatial-temporal mapping relationship between two successive thickness distributions with given $k.$ The trend of shape/thickness change in the input FTD snapshots will be selectively learned by the ConvLSTM and be applied to the feature maps in the fourth convolutional layer, forming the feature maps in the sixth convolutional layer. As a result, based on the

ConvLSTM, the high-density pixels in the sixth and eighth convolutional layers’ feature maps show more interest in the change in the shape/thickness of the water film, precisely the difference between the shape/thickness of the water film in future time steps and that in previous steps. The feature maps in the sixth convolutional layer are in low resolution and show the possible shape/thickness change area. In the eighth convolutional layer, some high-density areas in the sixth are abandoned, while others are strengthened by the optimized kernels in the eighth convolutional layer. Noteworthy here is that the sixth and eighth convolutional layers contain not only the feature maps with shape/thickness change information but also the feature maps with the shape/thickness information of input snapshots, like those in the fourth convolutional layer feature map. By combining the initial shape and the shape change of the water film, the output layer of ConvLSTM generates the final shape profile of the water film. A larger k leads to a larger shape change in the eighth convolutional layer, causing the water film to cover more area, as shown in Fig. 12. This indicates that the model is capable of predicting a longer future with a larger k value.

3. The robustness of the ConvLSTM-AutoEncoder against noisy inputs

The measurements from experimental techniques usually contain unavoidable noises. To assess the robustness of the proposed model against noises in the input sequence of FTDs, the Gaussian noise An was added to the input sequence of FTDs by $T_{noisy} = T + An,$ where n is the Gaussian noise, A is the amplitude of the noise, T is the original input sequence of FTDs, and T_{noisy} is the noisy input sequence of FTDs. Because our thickness data are measured in millimeters, the unit of the Gaussian noise amplitude A is a millimeter. Assume that the letter ϵ represents the experimental dataset’s uncertainty level, which is 0.02 mm. In this study, four noise levels, $A = 0.02$ mm (100% ϵ), $A = 0.05$ mm (250% ϵ), $A = 0.1$ mm (500% ϵ), and $A = 0.15$ mm (750% ϵ), are selected. The noisy FTD sequence of each tested case was fed into the corresponding trained model and evaluated the models’ performance using the AMAE metric. In Fig. 13, the AMAEs of the model with different noise amplitudes and different k are compared to evaluate the robustness of the model. It is worth noting that

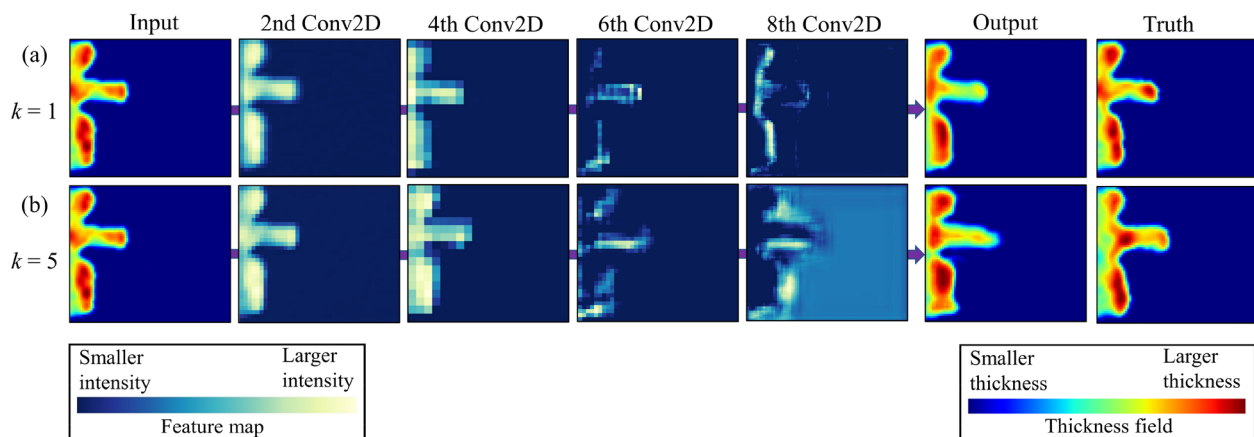


FIG. 12. Comparison of case #4’s latent feature maps before the ConvLSTM layer (i.e., the second and the fourth Conv2D layer) and after the ConvLSTM layer (i.e., the sixth and the eighth Conv2D layer). (a) The evolution of the latent feature maps in the model with $k = 1.$ (b) The evolution of the latent feature maps in the model with $k = 5.$

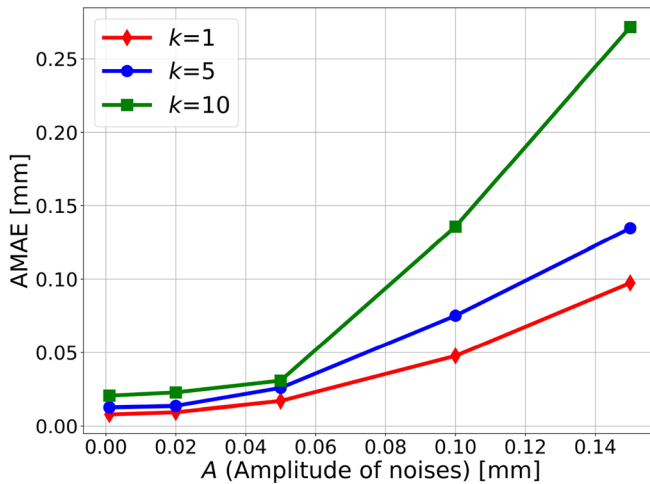


FIG. 13. Comparison of the robustness against noise of the ConvLSTM-AutoEncoder model with $k = 1, 5,$ and 10 .

the AMAEs grew slowly until the noise amplitude exceeded 0.05 mm. In other words, the proposed ConvLSTM-AutoEncoder model can effectively resist the noise introduced by the DIP system used to collect the experimental WDRWF data, which has an uncertainty measurement of about 0.02 mm. This assessment has the potential to serve as a benchmark for the application of the model in experimental research.

The predicted thickness distribution of two tested cases (i.e., case #1 and case #5) with different amplitudes of noisy inputs is shown in Fig. 14. For clarity, only the prediction with $k = 1$ is shown in Fig. 14, which corresponds to the red line in Fig. 13. When $A = 0.02$ mm, the predictions agree well with the experimental data. When $A = 0.05$ mm, the majority of the thickness distribution has no significant difference from the true data, except for one spot of non-zero thickness in the downstream region. It is caused by the noise in the

input FTDs. When $A = 0.10$ mm, some thin-thickness regions cannot be correctly predicted, as illustrated in Fig. 14(d).

IV. CONCLUSIONS

In the present study, machine learning methods are used for the first time to predict the time evolution of the front contact point (FCP) and film thickness distribution (FTD) of wind-driven runback water film (WDRWF) flows. For the FCP prediction, a comparative study is conducted to evaluate the performances of LightGBM and MLP models under different test conditions with varying freestream wind speeds and water flow rates. LightGBM is found to perform well in capturing intermittent behavior of film/rivulet head, while MLP is good at predicting smooth trajectories of WDRWF flows.

For FTD prediction, a computationally efficient deep neural network architecture called ConvLSTM-AutoEncoder is built to forecast the complicated spatial-temporal development of the multiphase WDRWF flow. The error of the model is found to be in the same order as the measurement uncertainty. The model is approved to be capable of accurately predicting the FTD in the next $k = 10$ time steps. The underlying mechanisms of the model are interpreted by analyzing the latent feature maps. The model is also demonstrated to be robust to real-world experimental applications. The model is generalizable to predict unobserved FTDs that have very different features from the training set. Based on the computational efficiency, generalizability, and robustness, the proposed model is an attractive alternative approach to traditional analytical and numerical approaches for real-world multiphase WDRWF flow, for which no exact numerical and analytical solutions exist. In the future, the model will be used to predict the evolution of ice accretion on airfoil surfaces generated by runback water film flow under glaze icing conditions. Based on the model’s prediction, an active flow control strategy will be explored in the future to switch on/off active ice protection systems in advance for effective aircraft icing mitigation. In addition, the model’s capability of predicting very large k (e.g., $k > 10$) will also be explored to expand the model’s scope of application.

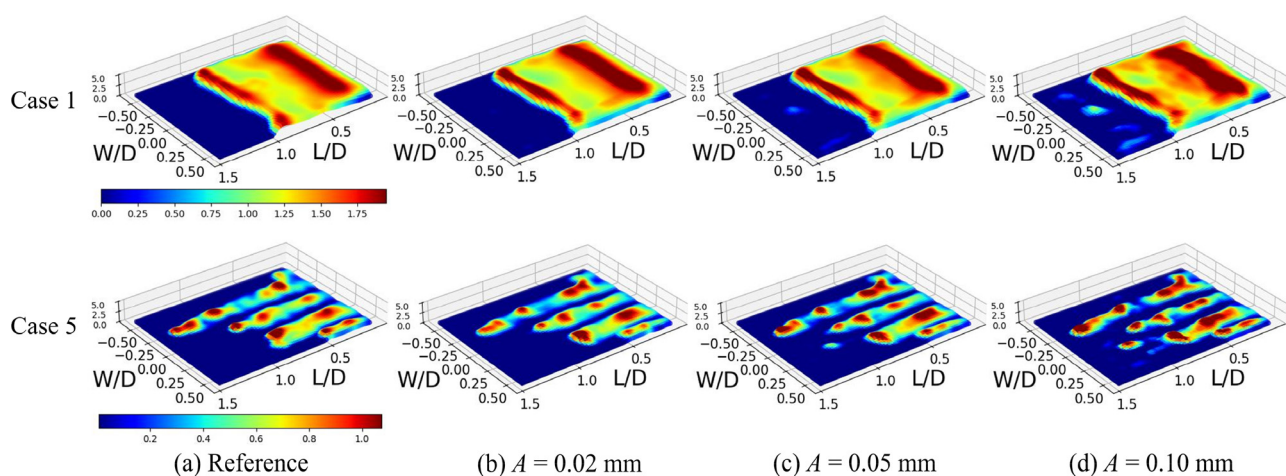


FIG. 14. Comparison of the outputs of the ConvLSTM-AutoEncoder model of cases #1 and case #5 with the noise magnitude of (b) $A = 0.02$, (c) $A = 0.05$, and (d) $A = 0.10$ mm. (a) The corresponding experimental data for reference, where the FTD of case #1 is taken at $t = 4.60$ s and that of case #5 is taken at $t = 0.87$ s.

03 October 2023 15:40:53

ACKNOWLEDGMENTS

The research work was partially supported by the Iowa Energy Center for Wind Turbine Icing Study under the IEC Competitive Grant No. 312350 and National Science Foundation (NSF) under Award Nos. CBET-1916380, CBET-1935363, TIP-2140489, and CBET-2313310.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Jincheng Wang: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Haiyang Hu:** Data curation (equal); Formal analysis (equal); Investigation (equal); Visualization (equal). **Ping He:** Conceptualization (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal). **Hui Hu:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹F. Caliskan and C. Hajiyev, “A review of in-flight detection and identification of aircraft icing and reconfigurable control,” *Prog. Aerosp. Sci.* **60**, 12–34 (2013).
- ²A. Heinrich *et al.*, *Aircraft Icing Handbook, Volume 1 of 3* (Gates Learjet Corporation, Wichita, KS, 1991), Vol. 1, p. ADA238040.
- ³Y. Cao, W. Tan, and Z. Wu, “Aircraft icing: An ongoing threat to aviation safety,” *Aerosp. Sci. Technol.* **75**, 353–385 (2018).
- ⁴M. B. Bragg, G. M. Gregorek, and J. D. Lee, “Airfoil aerodynamics in icing conditions,” *J. Aircr.* **23**, 76–81 (1986).
- ⁵C. Zhang and H. Liu, “Effect of drop size on the impact thermodynamics for supercooled large droplet in aircraft icing,” *Phys. Fluids* **28**, 62107 (2016).
- ⁶W. A. Olsen and E. N. Walker, “Experimental evidence for modifying the current physical model for ice accretion on aircraft surfaces,” Technical Report No. NASA-TM-87184 (NASA, 1986).
- ⁷Y. Liu and H. Hu, “An experimental investigation on the unsteady heat transfer process over an ice accreting airfoil surface,” *Int. J. Heat Mass Transfer* **122**, 707–718 (2018).
- ⁸S. P. Otta and A. P. Rothmayer, “Instability of stagnation line icing,” *Comput. Fluids* **38**, 273–283 (2009).
- ⁹A. P. Rothmayer and H. Hu, “Solutions for two-dimensional instabilities of ice surfaces uniformly wetted by thin films,” AIAA Paper No. AIAA 2012-3133, 2012.
- ¹⁰Y. Liu, K. Zhang, W. Tian, and H. Hu, “An experimental investigation on the dynamic ice accretion and unsteady heat transfer over an airfoil surface with embedded initial ice roughness,” *Int. J. Heat Mass Transfer* **146**, 118900 (2020).
- ¹¹Y. Liu, K. Zhang, W. Tian, and H. Hu, “An experimental study to characterize the effects of initial ice roughness on the wind-driven water runback over an airfoil surface,” *Int. J. Multiphase Flow* **126**, 103254 (2020).
- ¹²R. M. Waldman and H. Hu, “High-speed imaging to quantify transient ice accretion process over an airfoil,” *J. Aircr.* **53**, 369–377 (2016).
- ¹³R. J. Hansman, Jr. and S. R. Turnock, “Investigation of surface water behavior during glaze ice accretion,” *J. Aircr.* **26**, 140–147 (1989).
- ¹⁴Y. Cao, K. Yuan, and G. Li, “Effects of ice geometry on airfoil performance using neural networks prediction,” *Aircr. Eng. Aerosp. Technol.* **83**, 266–274 (2011).
- ¹⁵D. W. McCann, “NNICE—A neural network aircraft icing algorithm,” *Environ. Modell. Software* **20**, 1335–1342 (2005).
- ¹⁶E. Ogretim, W. Huesch, and A. Shinn, “Aircraft ice accretion prediction based on neural networks,” *J. Aircr.* **43**, 233–240 (2006).
- ¹⁷D. Yu *et al.*, “A multi-autoencoder fusion network for fast image prediction of aircraft ice accretion,” *Phys. Fluids* **34**, 76107 (2022).
- ¹⁸S. Yang, Y. Hou, Y. Shang, and X. Zhong, “BPNN and CNN-based AI modeling of spreading and icing pattern of a water droplet impact on a supercooled surface,” *AIP Adv.* **12**, 45209 (2022).
- ¹⁹I. K. Deo and R. Jaiman, “Predicting waves in fluids with deep neural network,” *Phys. Fluids* **34**, 67108 (2022).
- ²⁰H. Kagemoto, “Forecasting a water-surface wave train with artificial intelligence—A case study,” *Ocean Eng.* **207**, 107380 (2020).
- ²¹D. Sun *et al.*, “A deep learning-based bias correction method for predicting ocean surface waves in the Northwest Pacific Ocean,” *Geophys. Res. Lett.* **49**, e2022GL100916, <https://doi.org/10.1029/2022GL100916> (2022).
- ²²T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics (Springer, New York, 2001).
- ²³T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, 2016), pp. 785–794.
- ²⁴L. G. Ekelund, G. Johnsson, A. Melcher, and L. Orö, “Effects of cedilanid-D in combination with metoprolol on exercise tolerance and systolic time intervals in angina pectoris,” *Am. J. Cardiol.* **37**, 630 (1976).
- ²⁵S. Li, J. Qin, M. He, and R. Paoli, “Fast evaluation of aircraft icing severity using machine learning based on XGBoost,” *Aerospace* **7**, 36 (2020).
- ²⁶S. Li and R. Paoli, “Comparison of machine learning models for data-driven aircraft icing severity evaluation,” *J. Aerosp. Inf. Syst.* **18**, 973–977 (2021).
- ²⁷S. Li, J. Qin, and R. Paoli, “Data-driven machine learning model for aircraft icing severity evaluation,” *J. Aerosp. Inf. Syst.* **18**, 876–880 (2021).
- ²⁸F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural Comput.* **12**, 2451–2471 (2000).
- ²⁹A. T. Mohan and D. V. Gaitonde, “A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks,” [arXiv:1804.09269](https://arxiv.org/abs/1804.09269).
- ³⁰S. Pawar *et al.*, “A deep learning enabler for nonintrusive reduced order modeling of fluid flows,” *Phys. Fluids* **31**, 85101 (2019).
- ³¹Z. Deng, Y. Chen, Y. Liu, and K. C. Kim, “Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework,” *Phys. Fluids* **31**, 75108 (2019).
- ³²L. Zhu, W. Zhang, J. Kou, and Y. Liu, “Machine learning methods for turbulence modeling in subsonic flows around airfoils,” *Phys. Fluids* **31**, 15105 (2019).
- ³³Y. Li, J. Chang, Z. Wang, and C. Kong, “An efficient deep learning framework to reconstruct the flow field sequences of the supersonic cascade channel,” *Phys. Fluids* **33**, 56106 (2021).
- ³⁴R. Han, Y. Wang, Y. Zhang, and G. Chen, “A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network,” *Phys. Fluids* **31**, 127101 (2019).
- ³⁵J. Mi, X. Jin, and H. Li, “Cascade-Net for predicting cylinder wake at Reynolds numbers ranging from subcritical to supercritical regime,” *Phys. Fluids* **35**, 75132 (2023).
- ³⁶S. Laima, X. Zhou, X. Jin, D. Gao, and H. Li, “DeepTRNet: Time-resolved reconstruction of flow around a circular cylinder via spatiotemporal deep neural networks,” *Phys. Fluids* **35**, 15118 (2023).
- ³⁷X. Jin, S. Laima, W.-L. Chen, and H. Li, “Time-resolved reconstruction of flow field around a circular cylinder by recurrent neural networks based on non-

- time-resolved particle image velocimetry measurements,” *Exp. Fluids* **61**, 114 (2020).
- ³⁸K. Zhang and H. Hu, “An experimental study on the transient runback characteristics of wind-driven film/rivulet flows,” *Phys. Fluids* **33**, 112104 (2021).
- ³⁹K. B. Laskey, N. Xu, and C. H. Chen, “Propagation of delays in the national air-space system,” [arXiv:1206.6859](https://arxiv.org/abs/1206.6859).
- ⁴⁰F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” [arXiv:1201.0490](https://arxiv.org/abs/1201.0490).
- ⁴¹F. Rosenblatt, “The Perceptron—A perceiving and recognizing automaton,” Technical Report No. 85-460-1 (Cornell Aeronautical Laboratory, 1957).
- ⁴²D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature* **323**, 533–536 (1986).
- ⁴³E. Alpaydin, “Neural networks and deep learning,” in *Machine Learning* (Determination Press San Francisco, CA, 2021), Vol. 25.
- ⁴⁴D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in The 3rd International Conference on Learning Representations (ICLR), 2015.
- ⁴⁵A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems* edited by H. Wallach (Curran Associates, Inc., 2019), Vol. 32.
- ⁴⁶M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation* (USENIX Association, 2016), pp. 265–283.
- ⁴⁷X. Shi *et al.*, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems* (MIT Press, 2015), pp. 802–810.
- ⁴⁸A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM* **60**, 84–90 (2017).
- ⁴⁹V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning* (Omnipress, Madison, WI, 2010), pp. 807–814.
- ⁵⁰A. Senior and X. Lei, “Fine context, low rank, softplus deep neural networks for mobile speech recognition,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (IEEE, 2014).
- ⁵¹L. Vinet and A. Zhedanov, “A ‘missing’ family of classical orthogonal polynomials,” *J. Phys. A* **44**, 618 (2011).
- ⁵²R. M. Waldman, H. Li, and H. Hu, “An experimental investigation on the effects of surface wettability on water runback and ice accretion over an airfoil surface,” AIAA Paper No. AIAA 2016-3139, 2016.