

Article

Artificial Neural Network and Support Vector Regression for Predicting Turbulent Bursting in Bluff-Body Hydrodynamics

Anjan Samanta ¹  and Sankar Sarkar ^{2,*} 

¹ Department of Applied Statistics, Maulana Abul Kalam Azad University of Technology, Nadia 741249, West Bengal, India; anjan.samanta@makautwb.ac.in

² Physics and Applied Mathematics Unit, Indian Statistical Institute, Kolkata 700108, West Bengal, India

* Correspondence: sankar_s@isical.ac.in

Abstract

Machine learning prediction of turbulent bursting in near- and far-wake flow zones past two horizontal cylinders was studied in the present article. Based on the bursting dataset, two predictive models were constructed using Artificial Neural Networks (ANNs) and Support Vector Regression (SVR) with stress ratios as target values for each bursting event. After analyzing a number of plots, it was observed that the ANN and SVR models achieved satisfactory estimation accuracy, with minor overfitting specifically in the case of ANN models. By using deep learning for quadrant analysis and highlighting the adaptability of machine learning methods in open-channel turbulence, the current work should strengthen the understanding of bursting occurrences in bluff-body hydrodynamics.

Keywords: open channel flow; bursting; neural network; support vector machine

1. Introduction

Wake flow downstream of a wall-mounted bluff body has continued to be of interest to hydraulicians because it may be used in a variety of real-world hydraulic engineering environments, such as geophysical and environmental flow structures. The erosion of riverbeds brought on by hydraulic structures is mostly dependent on the wall-wake flow, which is the extended interrupted flow zone downstream of a wall-mounted obstacle. A number of studies on open-channel turbulence from 1960 to 1970 (Kline et al. [1]; Corino and Brodkey [2]; Grass [3]) showed that the flow pattern near the wall in a turbulent boundary layer is repetitive and takes the form of a quasi-cyclic process, known as the bursting process. Nonetheless, one of the most efficient and straightforward methods for comprehending turbulent bursting in two dimensions is the quadrant analysis which Lu and Willmarth proposed [4].

So far as the measurement and analysis techniques are concerned, several researchers tried over the past few decades to build the velocity profile and comprehend the turbulent structure in open-channel flow. Particle Image Velocimetry (PIV), Laser-Doppler anemometers, Acoustic Doppler velocimeters, and particle-tracking velocimeters were utilized in laboratory experiments and field research. These investigations validated the mathematical description of turbulence regimes and revealed qualitative patterns of turbulence in open-channel flow. However, in spite of taking sufficient precautions, there remain some locations where capturing the data by using any instrument becomes difficult, and sometimes there is a possibility of acquisition erroneous data due to the limitations of the measuring instruments, no matter how sophisticated they are. Given the shortcomings



Academic Editor: Giuseppe Pezzinga

Received: 24 May 2026

Revised: 18 June 2026

Accepted: 23 June 2026

Published: 26 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

found in turbulence measurement, numerous scholars have underscored the necessity of enhancing laboratory and field investigations. The creation of different approaches is also necessary in order to properly handle the simulation of complex circumstances in open channels. Researchers from a wide range of scientific and engineering fields are currently becoming more interested in artificial neural networks, or ANNs. Additionally, ANNs have shown promise in the fields of hydrology and water resources. For instance, Chang et al. [5] assessed the viability of utilizing artificial neural networks to simulate the mean velocity and turbulence intensity profiles for steep open-channel flows across a smooth boundary. The results showed that while the log law and Reynolds Stress Model (RSM) were less successful at simulating the velocity profiles close to the side wall, ANN could accurately replicate the velocity profiles. Yang and Chen [6] used ANN to investigate possible hazards from the standpoint of flow monitoring in big open-channel junctions. The outcomes demonstrated the ANN's ability to dependably and precisely reproduce the mean velocity profiles. Ma et al. [7] employed direct numerical modeling of bubbly multiphase flows with neural networks to derive closure terms for a basic average-flow model. After being trained on a dataset from a single simulation, the ANN was used to model how different initial conditions will evolve. All things considered, the generated model did an adequate job of predicting how the different initial conditions will evolve. The ability of ANN models to represent and model the velocity distributions of coupled open-channel flows was examined by Sun et al. [8]. The majority of the uncertainty, according to their results, came from the initialization of the ANN model parameters. In order to lessen the primary source of uncertainty, an ANN model training approach was created. Models were trained over a number of runs using random model parameter initializations, and the model that performs the best was chosen. Drikakis and Sofos [9] highlighted algorithmic issues, reviewed the existing literature on machine learning (ML) and deep learning for fluid dynamics, and talked about possible future paths. According to them, it is crucial that AI (Artificial Intelligence) and ML techniques go beyond basic prediction tasks and delve deeply into interpretability and causality. If the output follows physical rules, then its potential impact would be quite significant. In the instance of coupled open-channel flow, Yang et al. [10] assessed the suitability of the ANN for simulating velocity profiles and velocity contours and calculating the discharges correspondingly. The findings showed that the ANN can accurately forecast the discharges for the conditions under investigation and could simulate the velocity profiles. In the paper by Yuhong et al. [11], two input parameters—the Reynolds number and the relative roughness—were used to set up a three-layer ANN model to forecast the friction factors of an open-channel flow. The successful implementation showed that ANN modeling might be a practical and efficient tool in engineering practice, and it could be used to assess classic hydraulic problems, the majority of which were based on laboratory experiments. Fang et al. [12] focused on modeling the Reynolds stress closure for the particular case of turbulent channel flow and suggested three changes to a conventional neural network to take the anisotropy tensor's no-slip boundary condition, the Reynolds number dependence, and spatial non-locality into consideration. When the updated models were trained and evaluated on channel flow in various Reynolds numbers, it was demonstrated that they yield higher predictive accuracy than the conventional neural network.

Beyond the configuration taken into consideration in this study, machine learning has been widely adopted in industrial fluid mechanics. Data-driven models provide a computationally efficient method of prediction, optimization, and control for flows that are too costly or complicated to handle with high-fidelity simulation alone [13,14]. Deep learning models have been used to reconstruct near-wall flow fields in wall-bounded turbulence and to guide active control strategies aimed at achieving objectives like improved

mixing and drag reduction [15]. One application that is especially pertinent is pipeline transport. Friction drag is still a persistent source of energy loss in the transportation of oil and natural gas through pipelines. In order to support turbulence-based control strategies for drag reduction, recent studies have used deep-learning models to predict turbulent pipe-flow fields that generalize across operating conditions [16]. The same reasoning holds true for rotating fluid machinery. Machine learning is increasingly being used for the design optimization, performance prediction, and fault diagnosis of pumps, hydraulic and wind turbines, compressors, and torque converters, which are essential to industries from aerospace and petrochemicals to water conservation and medical engineering [17]. This is demonstrated by centrifugal and multistage pumps, where conventional surrogates for estimating efficiency, head, and pressure pulsation have been largely replaced by surrogate models built from neural networks, support vector regression, and Gaussian-process techniques [18]. Therefore, the present approach is a component of a larger and rapidly developing initiative to incorporate data-driven methods into industrial fluid machinery.

The aforementioned discussion indicates that a lot of research has been carried out on both the development of machine learning models to analyze fundamental turbulent flow characteristics like velocity and intensity as well as the experimental investigation of turbulent bursting in open-channel flow; however, there is no study in the literature that compares experimental bursting results with machine learning predictions. Sarkar et al. [19] demonstrated how flows are affected in the presence of two cylinders, ranging from fundamental turbulent aspects like Reynolds Shear Stress (RSS) and intensities to complex features like TKE dissipation rate and budget. Furthermore, Samanta et al. [20,21] offered a number of advanced evaluations in the same experimental conditions, such as correlation coefficient, stress anisotropy, and turbulent bursting. Thus, this paper concentrates on estimations of turbulent bursting occurrences in the near-wake and far-wake zones downstream of two horizontal cylinders in order to expand on our earlier work. Based on the bursting dataset, two widely accepted machine learning algorithms, Artificial Neural Network (ANN) and Support Vector Regression (SVR), have been used for this purpose in order to construct two predictive models, which will be considered as an additional method of experimental investigation. Furthermore, it is prudent to acknowledge the very chaotic character of open-channel flow. Therefore, our current findings should be strengthened by the validation of our findings incorporating the ANN and SVR. Researchers in several domains, such as computer scientists and data analytics, should be motivated by the addition of ANN and SVR.

2. Experimental Details

The analysis of the present study is based on the experimental data of Sarkar et al. [19] and Samanta et al. [20], where detailed description of the open-channel setup and other relevant information are provided. The experiment was carried out in the Fluvial Mechanics Laboratory of the Indian Statistical Institute in Kolkata, India, in a rectangular-shaped flume of length 20 m, width 0.5 m, and height 0.5 m with transparent Perspex sidewalls, which is depicted in Figure 1. The working medium in this study is water. The flow was supplied and recirculated by two centrifugal pumps; furthermore, the bed was made rough by uniformly pasting gravel of median size $d = 2.49$ mm at a streamwise slope $S = 3 \times 10^{-4}$. The relevant flow parameters are as follows: incoming flow depth $h = 0.25$ m, and depth-averaged velocity $U_0 = 0.44$ ms⁻¹. Two horizontal cylinders of diameter $D = 0.038$ m were placed one above the other with a spacing equal to one cylinder diameter. The flow was subcritical and turbulent, with Froude number $Fr = 0.28$ and Reynolds number $Re \approx 440,000$. The instantaneous velocity component data were collected with an Acoustic Doppler Velocimeter (ADV) named Vectrino plus through the centerline of the flume.

The Vectrino system was implemented with a sampling rate of 100 Hz and an acoustic frequency of 10 MHz. In order to prevent the bottom wall from affecting the sampling volume, we commenced data collection just 3 mm above the gravel bed. We subsequently raised the sampling height by 2 mm. The data were gathered at each location for 240 s in order to obtain a time-independent time-average velocity profile. The uncertainty analysis, which was reported in detail in our previous works, indicated that velocity components maintained standard deviations below 0.5 cm/s and the average maximum error was under 5%. Since the models are trained on time-averaged quantities from a 240 s record, the influence of this small measurement uncertainty on the predictions is expected to be limited.



Figure 1. Photograph of the experimental setup.

Spikes often interfere with the gathered data because of the interaction between the incident and reflected pulses. The data were filtered using the acceleration thresholding technique, a spike removal method that could separate and replace the spikes in two phases. However, it was discovered that the Kolmogorov “ $-5/3$ scaling-law” for the velocity power spectra $F_{ii}(f)$ in the inertial subrange, where f is the frequency, was satisfied by the threshold values (1 to 1.5) for data cleaning. For example, Figure 2 displays the $F_{ii}(f)$ at a height of $z = 2.75D$ and a streamwise distance of $x = 1D$ following the spike elimination. The filtered signals of $F_{ii}(f)$ -curves are consistent with the “ $-5/3$ scaling-law” in the inertial subrange for $f \geq 0.5$ Hz.

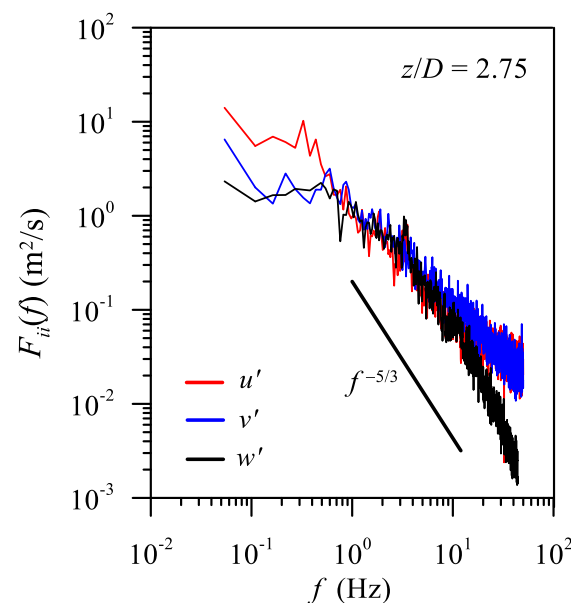


Figure 2. Power spectra density function of velocity after despiking, $F_{ii}(f)$ at a vertical distance of $z = 2.75D$ and a streamwise distance of $x = 1D$.

3. Artificial Neural Network (ANN) Formulation

Inspired by biological neuron processing, neural networks use artificial neurons with a very simple connective structure to execute computations akin to those of the brain. A number of neural networks, including the multi-layer perceptron neural network, the self-organizing network, the fuzzy neural network, and the radial basis function neural network, were developed after McCulloch and Pitts [22] developed the first neural network. The Multi-Layer Perceptron (MLP) neural network is a widely used neural network type that has been successfully applied for adaptive identification of diverse non-linear processes. An input layer, one or more hidden layers, and an output layer are the various layers of nodes (neurons) comprising a Multi-Layer Perceptron (MLP), a kind of feedforward artificial neural network. Every node in one layer is linked to every node in the other layer, and each connection has a weight attached to it. MLPs are trained using supervised learning, often with backpropagation and gradient descent-based optimization algorithms [23].

Figure 3 schematically shows the working procedure of a MLP neural network. There are three layers in it: input, hidden, and output. Numerous neurons, the basic building blocks of the network, are present in each layer. The input vector containing the training patterns is sent to the input layer. The way the input variables interact is represented by the hidden layer. A result is transported to output vectors via the output layer.

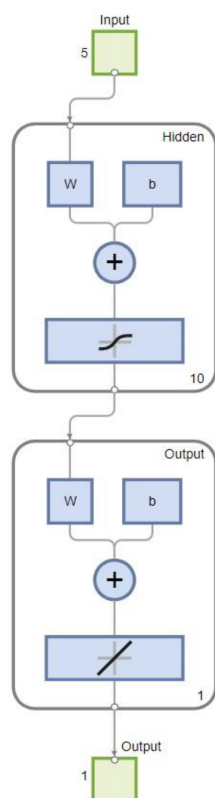


Figure 3. Schematic diagram of ANN.

- w represents the weight associated with each input connection to a neuron.
- b denotes the bias of the neuron, which acts as a constant threshold value.
- The summation operator (+) then adds up the product of each input and its respective weight, along with the bias term.

This combined value becomes the input to the neuron's transfer function, by which the learning process moves from the input layer to the hidden layer and finally to the output layer. Trial-and-error methodology is used to identify the number of neurons on the hidden layer. Increasing the count of neurons gradually continues until the ANN model

performance meets the desired level or until adding more neurons does not significantly improve the performance. Here, the optimal number of hidden layers is found out to be 10. To move weighed sums from neurons to neuron outputs, transfer functions that model non-linear relations are required. The output layer uses a linear transfer function, whereas the hidden layer uses a logistic-sigmoid transfer function, described as follows [24]:

$$f(t) = t \quad (1)$$

$$g(t) = \frac{1}{1 + e^{-t}} \quad (2)$$

Here, the Levenberg–Marquardt (LM) training technique is applied to determine precise predictions by an iterative adjustment in bias and weight. The algorithm is intended to generate the best result as soon as the least mean squared error is obtained. Numerous earlier studies have demonstrated the effectiveness of the LM approach in training ANN models [23,25–27]. If not specified, all accessible data will be divided into three sub-datasets for the purposes of testing, validating, and training artificial neural networks (ANNs). The training process is stopped when the ANN model's validation performance does not increase, and this decision is made based on the validation dataset.

4. Support Vector Machine and Regression (SVM and SVR) Formulation

On the basis of statistical learning theory, Vapnik [28] created the reliable and effective algorithm known as Support Vector Regression (SVR). It gained popularity as a result of its effective use in regression and classification jobs to obtain the lowest possible regression error, particularly for time series forecasting [29]. As a result, researchers that use SVR for time series forecasting, mostly in the water resource field, typically ignore margin setting options that modify cost function to obtain lower RMSE [30]. The efficacy of the non-linear SVR is contingent upon the margin constant parameter C , the ε -insensitive loss function, and γ [31]. Changing one of them has an impact on the other as they are highly interconnected. SVR maps the input x onto an m -dimensional feature space using a fixed (non-linear) mapping process before building a linear model in this feature space. Using a non-linear function $\phi : x \rightarrow f$ to translate our data from the input space X to a feature space F is the crude method of creating a non-linear classifier from a linear classifier. The discriminating function in space F is as follows:

$$f(x) = w^T \phi(x) + b \quad (3)$$

Which subsequently takes the following form:

$$f(x) = \sum \alpha_i k(x, x_i) + b \quad (4)$$

where α_i is the coefficient of linear combination of input vectors, and $k(x, x_i)$ is known as the kernel function. Since SVM has a large number of kernel functions, choosing an appropriate kernel function is a research question as well. Nonetheless, there are a few widely used kernel functions as follows:

- Linear Kernel: $k(x, y) = x^T y$.
- Polynomial Kernel: $k(x, y) = (\gamma x^T y + r)^d$, d being degree of the polynomial.
- Gaussian Kernel: $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$.

It is commonly recognized that the kernel parameters and the model parameters C , γ , and r must be adjusted properly for good SVM generalization performance (accuracy in prediction) to be achieved. The predictive (regression) model complexity is determined

by the decisions made by C , γ , and r . Due to the SVM model's dependence on all three parameters for complexity and, consequently, generalization performance, the challenge of optimum parameter selection is made significantly harder. Exploring the hyperparameter space manually is one way to accomplish this, but it takes time and raises the risk of producing predictions that are off. Therefore, it is possible to optimize parameters using a number of techniques that have been published in the literature [32,33]. Nevertheless, in our instance, manual search was sufficient to yield an ideal set of hyperparameters with a respectably low MSE.

5. Computational Methodology

ANN and SVR algorithms have been implemented using MATLAB software (MATLAB R2024a). There is one target variable, five features, and 504 observations in the dataset that was used to train the model. Streamwise distance (x/D), vertical distance (z/D), and average velocity components ($\bar{u}, \bar{v}, \bar{w}$) are being taken as input features, whereas stress ratio data from all four events ($S_{i,0}$) have been taken into consideration one by one for the output in order to build four different prediction models. The four events, namely, outward interactions (Q_1 event at quadrant I), ejections (Q_2 event at quadrant II), inward interactions (Q_3 event at quadrant III), and sweeps (Q_4 event at quadrant IV), were selected as response variables due to their significance in characterizing the overall system behavior and capturing variations in the underlying process dynamics. Predicting these responses enables the development of reliable data-driven models capable of estimating system performance under different operating conditions without the need for repeated experimental or computational evaluations. In this study, ANN and SVR models were employed to learn the complex relationships between the input parameters and the corresponding event responses. The prediction of these events is particularly important for improving computational efficiency, reducing analysis time, and facilitating rapid assessment of system behavior. Furthermore, accurate prediction of Q_1 – Q_4 provides a practical framework for future optimization and decision-making applications using machine learning approaches.

Readers may find a description of the dataset taking $S_{2,0}$ as output in Table 1. In the instance of ANN modeling, the whole set of data was randomly divided into three portions: 70% for training, 15% for validation, and 15% for testing. However, in the case of SVR, data are divided into training and testing segments at a ratio of 7 to 3. While designing the SVR model, several kernel functions, including Gaussian, radial, linear, and polynomial, were used; the Gaussian function performed better than the others. Table 2 contains all the hyperparameter information. To compare the SVR and ANN models' performances equitably, the same dataset that was used to train the SVR model was also used to train the ANN model, likewise with regard to the test dataset.

The creation of the aforementioned network by itself does not meet the criteria needed to qualify as an efficient prediction model. It is predicated on the accuracy report of the model, for which a variety of quantifiers are accessible in the literature. These include Mean Squared Error (MSE), Coefficient of Determination (R), and Margin of Deviation (MoD). When the model is being trained, validated, and tested, these parameter settings—which are described below—will guarantee the accuracy of the stress ratio prediction.

$$MSE = \frac{1}{n} \sum \left(Y_{ANN/SVR}^i - Y_{Exp}^i \right)^2 \quad (5)$$

$$R = 1 - \frac{\sum \left(Y_{ANN/SVR}^i - Y_{Exp}^i \right)^2}{\sum \left(Y_{Exp}^i \right)^2} \quad (6)$$

$$MoD = \frac{Y_{ANN/SVR}^i - Y_{Exp}^i}{Y_{Exp}^i} \times 100 \quad (7)$$

Table 1. Brief description of the data used for the training, validation, and testing of the ANN and SVR models taking $S_{2,0}$ as a response.

z/D	x/D	$\bar{u}$	$\bar{v}$	$\bar{w}$	$S_{2,0}$
0.052632	−2	0.065533	0.004847	−0.00024	1.289756
0.105263	−2	0.158147	0.007837	0.001427	1.116914
0.157895	−2	0.186979	0.013238	−0.00015	0.887275
0.210526	−2	0.217836	0.015306	−0.00023	0.787628
0.263158	−2	0.230588	0.016854	0.000187	0.794145
0.315789	−2	0.243035	0.013036	0.000237	0.800398
0.368421	−2	0.256859	0.021642	6.20×10^{-5}	0.954617
0.421053	−2	0.273217	0.025325	−0.00068	0.921584
0.473684	−2	0.276254	0.021658	−0.00037	0.785573
⋮	⋮	⋮	⋮	⋮	⋮
2.894737	10	0.315072	0.028404	−0.012	0.886
3.026316	10	0.327843	0.028377	−0.01333	0.94197
3.157895	10	0.326917	0.043679	−0.01365	0.895
3.289474	10	0.335137	0.021997	−0.01059	0.911414
3.421053	10	0.349977	0.032986	−0.01316	0.86194

Table 2. Details of the hyperparameters used in the SVR model.

SVR Hyperparameters Used	Values
C	150
ϵ	0.01
Kernel function	Gaussian

6. Results and Discussion

One of the data sources utilized to examine the ANN models' learning and training processes involves analyzing the model training performance visualizations. Figure 4 shows the four model performances throughout the processes of testing, validation, and training. The graph displaying the calculated MSE values for the provided dataset makes it clear that the MSE values are high at the start of the training phase. In the case of the $S_{2,0}$ model, after the second epoch, the training and validation errors are becoming asymptotic to the epoch axis, which indicates the model is starting to converge, and the model's performance has been improved up to the ninth epoch, or iteration, with a raised profile for validation being visible henceforth. After the ninth epoch, there are six consecutive validation failures observed, which means the iteration ends to provide the best-performing ninth epoch with the lowest MSE value. Here, validation fails occur as the model's performance on the validation dataset starts to degrade or does not improve after the best performing epoch. These findings are indeed supported in Figure 5b, where the gradient of the error function and the failure of validation serve to illustrate the training state. It shows that the error gradient is minimal at the ninth epoch, after which it depicts a tendency for gradual increment, leading towards consecutive validation failures. The model performance for

the rest of the stress ratios can be explained in a similar manner, which provides the best performing epoch as 40, 25, and 77 for the $S_{1,0}$, $S_{3,0}$, and $S_{4,0}$ models, respectively.

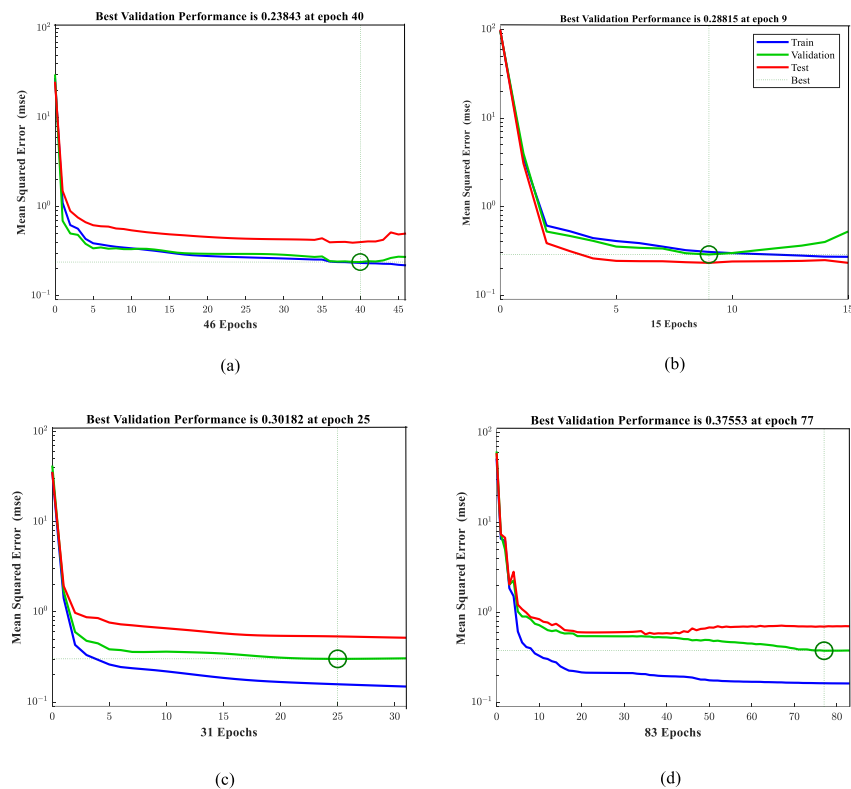


Figure 4. Model performance throughout the processes of testing, validation, and training for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

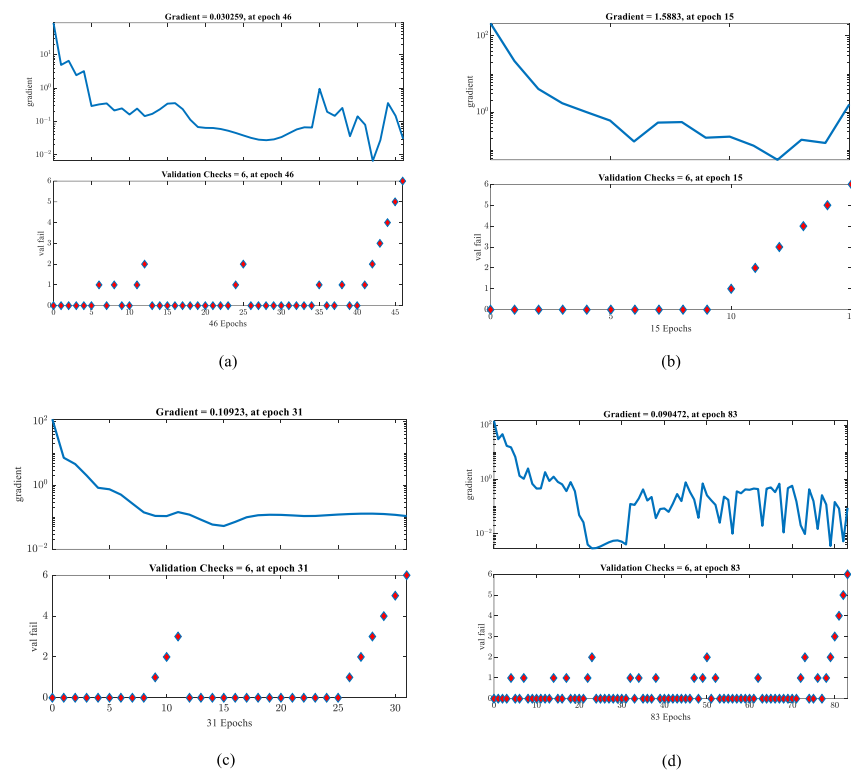


Figure 5. Gradient of the error function and the failure of validation in the ANN model for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

An additional technique to evaluate the training performance of ANNs is the examination of error histograms, which display the errors made during the processes of training, validation, and testing. The MLP neural network's error histogram, displayed in Figure 6, displays errors committed during training for all the stress ratio models. Based on the findings of the error histograms, a small amount of error is made at every level of the ANN model. The zero error line, shown by the orange line on the data histogram, indicates the location of the largest concentration of errors. The errors as graphically indicated follow normal distribution with the mean coinciding with the zero error line. This indicates that the majority of the predictions have errors close to zero, meaning that, on average, the ANN model is making predictions very near the true values. The vertical line at zero error essentially represents instances where the predicted values match the true values. The error distribution is consistent across the training, validation, and test datasets, suggesting good generalization.

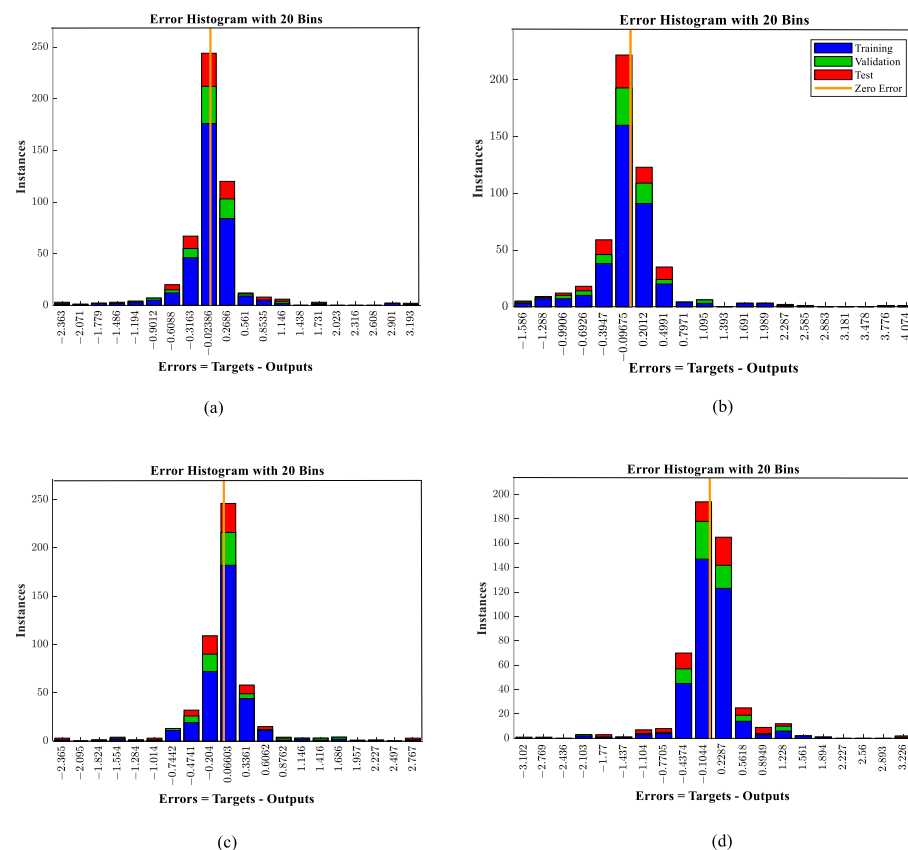


Figure 6. Error histogram in the ANN model for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

The MSE values for each of the data used to train both ANN and SVR models are shown in Figure 7 using vertical bars for all stress ratio models. When the models are subjected to a fresh dataset, it is evident that the projected MSE values produce pretty low results, suggesting appropriate training and reliable prediction. In the case of both models, the initial MSE values were quite large, indicating modeling complexity with the dataset from the near-wake zone. The Margin of Deviation (*MoD*) values for the ANN and SVR models are represented by the data points in Figure 8. It is found that these values tend to be low and close to the line of zero deviation, with average deviation values 2.4% and 0.84%; 2.83% and 7.06%; 7.08% and 7.01%; and 2.23% and 4.35% for the $S_{i,0}$, $i = 1 \dots 4$ models, respectively. It indicates that in terms of deviation both ANN and SVM performances are excellent for prediction task as their deviations are within 10%. Again, near-wake zone models are struggling to produce a good fit with the large *MoD*

values, supported by the findings in Figure 7. It is worth analyzing why both models exhibit their largest deviations in the near-wake zone. This region is physically distinctive from the rest of the flow because it is dominated by flow reversal, recirculation, and strong bursting events, which lead to sharply peaked and high-magnitude stress ratios, in contrast to the uninterrupted upstream flow and the gradually recovering far-wake zone (see Sarkar et al. [19] and Samanta et al. [21] for the detailed turbulence characterization of this configuration). Consequently, the near-wake zone produces a small number of extreme, high-variance samples to the dataset which the models will treat as outliers. Since both the ANN and SVR are trained to minimize overall error, they fit through the data with moderate variances, and they underrepresent these outliers located in the near-wake zone, due to which such large prediction errors are observed in this specified zone only.

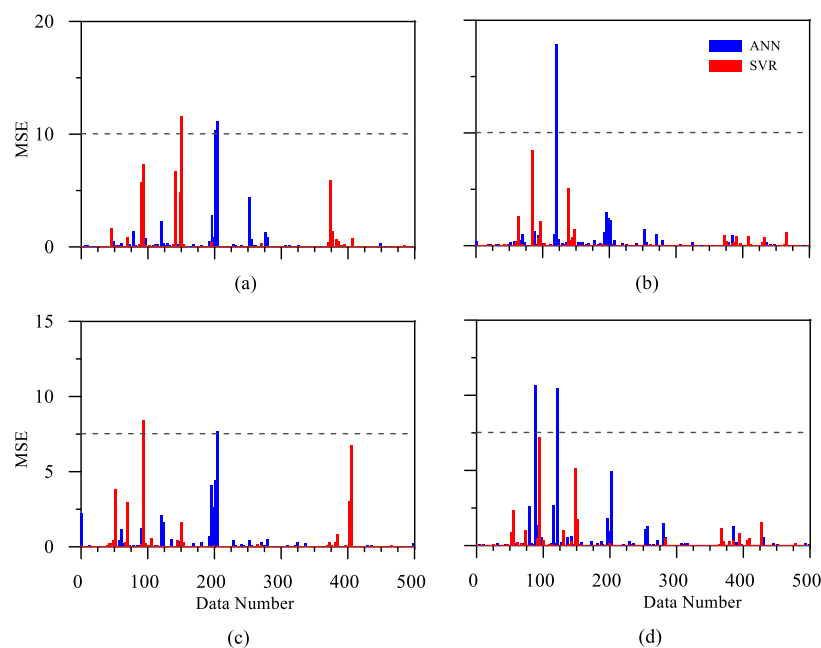


Figure 7. The MSE values for each of the data used to train both ANN and SVR for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

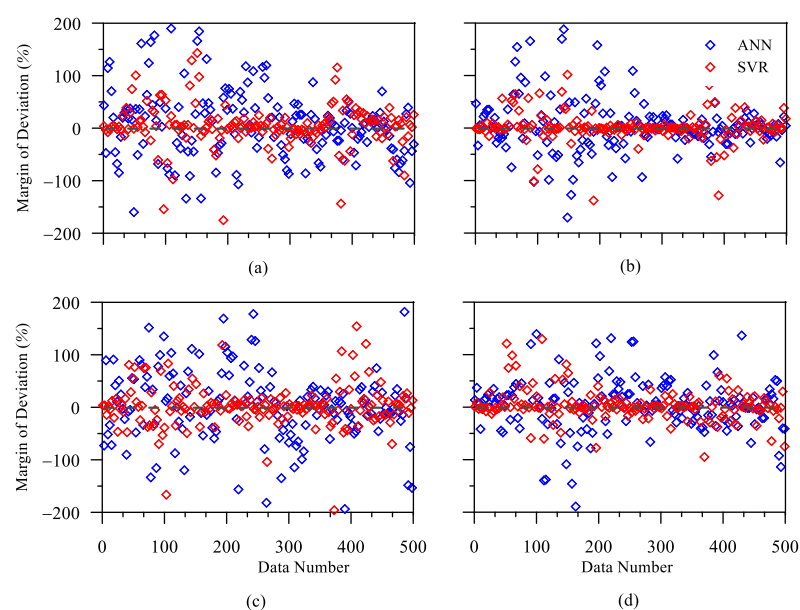
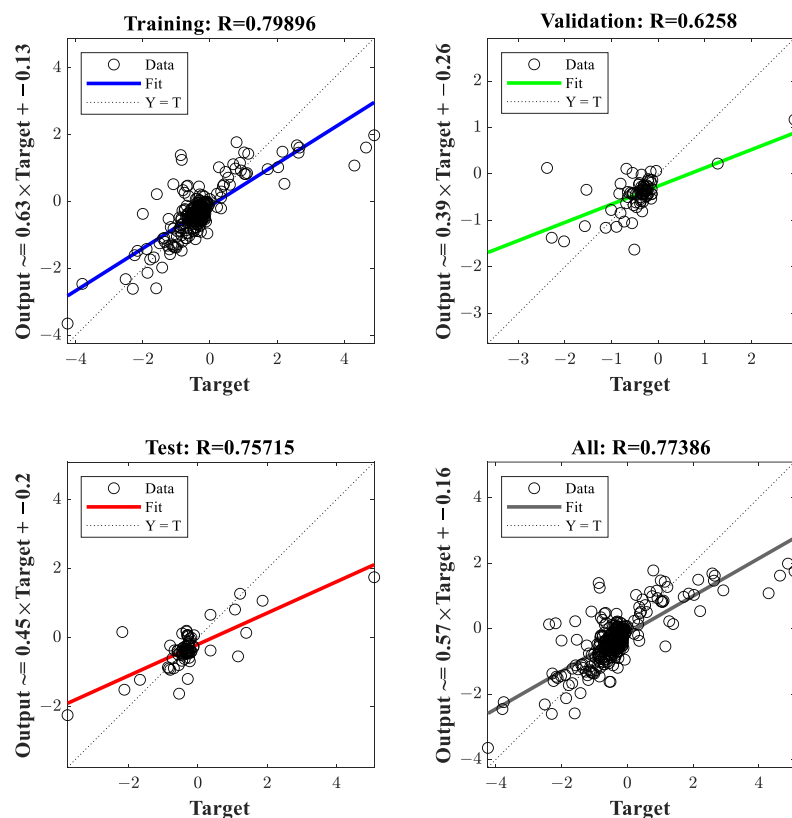


Figure 8. The Margin of Deviation (*MoD*) values for the ANN and SVR for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

At every step, the degree to which the projected data match the original data well is evaluated using the coefficient of determination (R). A model's quality of agreement is essentially shown by an R that is closer to 1. In all ANN models, the R values from training and testing, respectively, are 0.79896 and 0.75715; 0.71405 and 0.65682; 0.81043 and 0.60526; and 0.82416 and 0.63197, for the $S_{i,0}$, $i = 1 \dots 4$ models, as shown in Figure 9, where most of data plotted are fallen onto the line. This essentially indicates that the ANN model has high prediction accuracy. It is observed that, in most cases, the testing accuracy is relatively smaller than the training accuracy, which is acceptable as it is conventional for the model to underperform a little under the exposure to a new dataset. With R values for training and testing 0.6185 and 0.6613; 0.6049 and 0.6119; 0.6132 and 0.6208; and 0.6455 and 0.6363, respectively, for the SVR model, Figure 10 shows a similar feature. As observed for the $S_{1,0}$ model, ANN performs fairly well compared to SVR in terms of accuracy. For the $S_{2,0}$ model, ANN is found out to be more efficient for predictive tasks than SVR. But it is more challenging to conclude which is the best performing model in the case of the $S_{3,0}$ and $S_{4,0}$ models. If the overall accuracy is taken into account, it seems that the ANN model is better performing than SVR; however, there are significant drops in testing accuracy being observed for ANN models. A significant difference between the training accuracy (0.81, 0.82) and testing accuracy (0.60, 0.63) typically indicates overfitting. Overfitting occurs when a model learns the training data too well, including its noise and details, resulting in poor generalization to new, unseen data. Hence, to obtain a more reliable prediction accuracy for unknown data and to avoid overfitting, SVR is recommended for stress ratio models $S_{3,0}$ and $S_{4,0}$.



(a)

Figure 9. Cont.

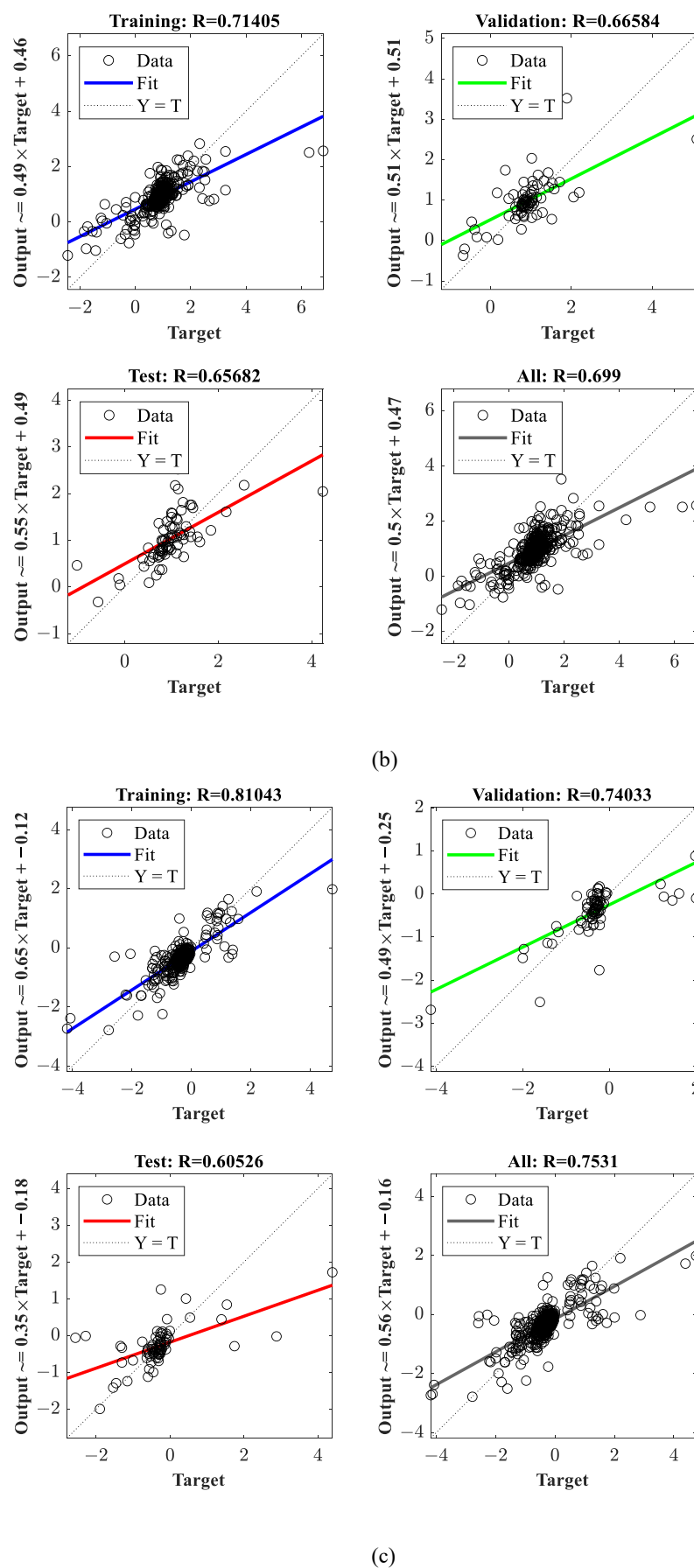
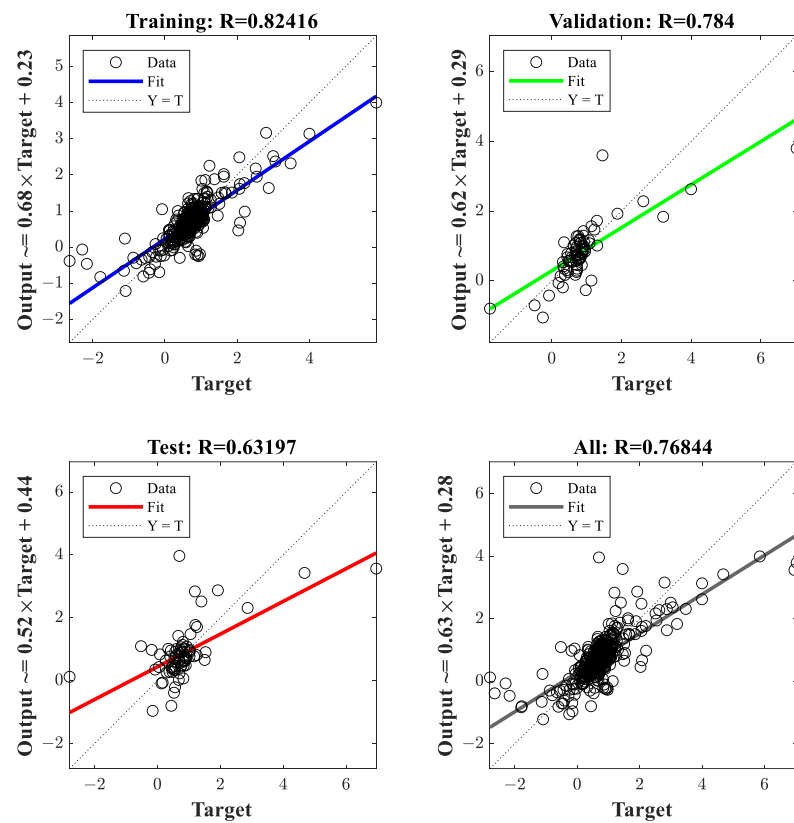


Figure 9. Cont.



(d)

Figure 9. Cross plots illustrating the second quadrant's actual and ANN-generated values in the following four phases: training, validation, testing, and overall for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

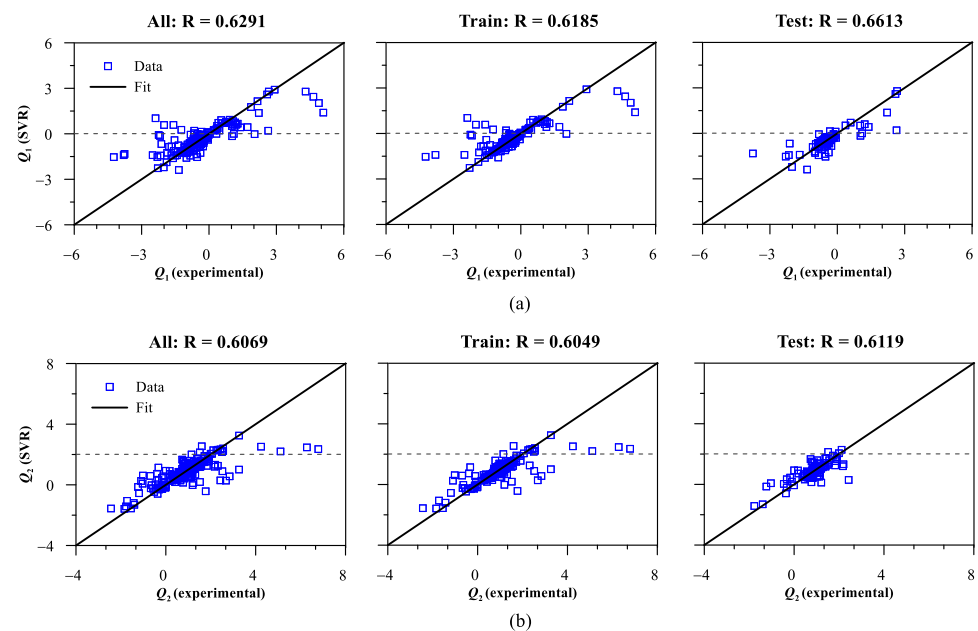


Figure 10. Cont.

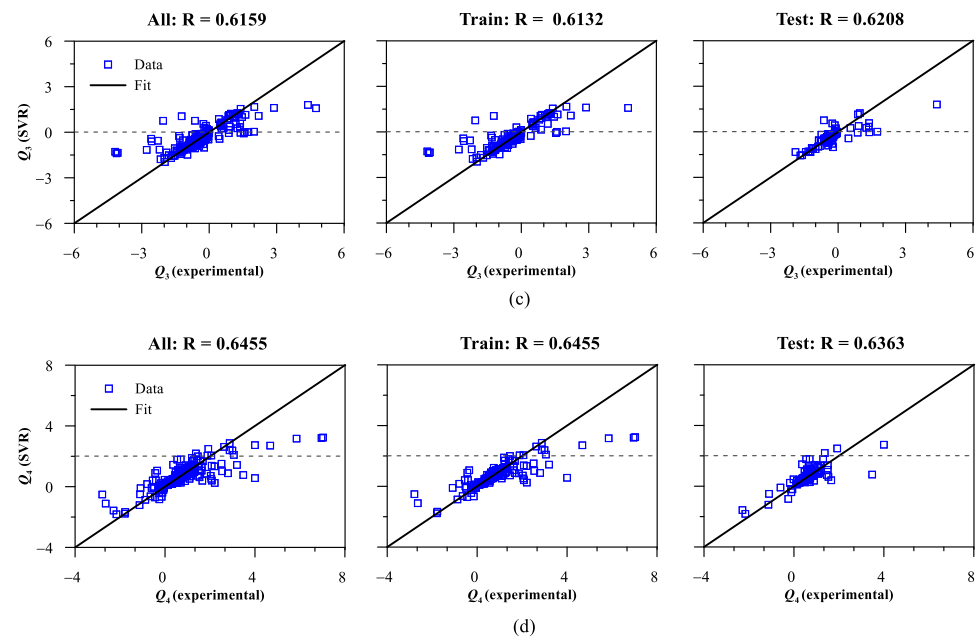


Figure 10. Cross plots illustrating the second quadrant's actual and SVR-generated values in the following three phases: overall, training, and testing for (a) $S_{1,0}$, (b) $S_{2,0}$, (c) $S_{3,0}$, and (d) $S_{4,0}$ as responses.

7. Conclusions

This study aims to illustrate a predictive model building approach towards the estimation of turbulent bursting in an open-channel flow with two horizontal cylinders acting as bluff bodies. In order to present an alternative approach of bursting evaluation without experimental setup, two machine learning algorithms, Artificial Neural Network and Support Vector Regression, have been modeled and discussed. The numerical values of the stress ratios for all events generated by the ANN or SVR models and their experimental values have minimal deviations according to the *MoD* results. Due to the overfitting issues observed in the case of ANN, especially for models with response $S_{3,0}$ and $S_{4,0}$, SVR is recommended for the said models. Upon comparison of all three stress ratio values obtained from experimental, ANN, and SVR models, it is found that the models were unable to accurately anticipate in a few positions, particularly at the near-wake zone of the bluff-bodies. In summary, the recommended ANN and SVR models have been built in a way that offers an additional approach for predicting stress ratios for all bursting events with high accuracy and acceptable error rates, especially at those positions where it is not feasible to perform experimental data collection.

Collecting highly accurate experimental data for open-channel flow requires sophisticated measurement techniques and equipment such as Particle Image Velocimetry (PIV), which can be costly and time-consuming. As a result, there may be limited quantities of high-quality experimental data available for training ML models, potentially leading to issues such as overfitting. Also, it is important to remember that because the models presented here were developed using the data gathered from the experimental setup, they are only applicable to wall-wake flow past two horizontal cylinders. As such, we do not expect the models to have particularly high prediction accuracy when employed with bluff bodies other than cylinders. Therefore, our next aim would be to integrate machine learning with physics-based models [34], which offers a potential solution to enhance predictive accuracy while maintaining physical interpretability. Lastly, Ensemble learning techniques [35], such as model averaging or stacking, will be employed to mitigate the

limitations of individual models and improve overall prediction accuracy across diverse bluff bodies and flow conditions.

Author Contributions: Conceptualization, A.S. and S.S.; methodology, A.S. and S.S.; software, A.S.; validation, A.S. and S.S.; formal analysis, A.S. and S.S.; investigation, A.S. and S.S.; resources, A.S. and S.S.; data curation, A.S. and S.S.; writing—original draft preparation, A.S.; writing—review and editing, S.S.; visualization, A.S. and S.S.; supervision, S.S.; project administration, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Engineering Research Board, Government of India, grant number CRG/2023/008825.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Acknowledgments: The authors acknowledge the constructive comments provided by the reviewers that helped us to improve the overall quality of the manuscript significantly.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kline, S.J.; Reynolds, W.C.; Schraub, F.A.; Runstadler, P.W. The structure of turbulent boundary layers. *J. Fluid Mech.* **1967**, *30*, 741–773. [\[CrossRef\]](#)
2. Corino, E.R.; Brodkey, R.S. A visual investigation of the wall region in turbulent flow. *J. Fluid Mech.* **1969**, *37*, 1–30. [\[CrossRef\]](#)
3. Grass, A.J. Structural features of turbulent flow over smooth and rough boundaries. *J. Fluid Mech.* **1971**, *50*, 233–255. [\[CrossRef\]](#)
4. Lu, S.; Willmarth, W. Measurements of the structure of the Reynolds stress in a turbulent boundary layer. *J. Fluid Mech.* **1973**, *60*, 481–511. [\[CrossRef\]](#)
5. Chang, F.J.; Yang, H.C.; Lu, J.Y.; Hong, J.H. Neural network modelling for mean velocity and turbulence intensities of steep channel flows. *Hydrol. Process. Int. J.* **2008**, *22*, 265–274.
6. Yang, H.C.; Chen, C.W. Potential hazard analysis from the viewpoint of flow measurement in large open-channel junctions. *Nat. Hazards* **2012**, *61*, 803–813.
7. Ma, M.; Lu, J.; Tryggvason, G. Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system. *Phys. Fluids* **2015**, *27*, 092101. [\[CrossRef\]](#)
8. Sun, S.; Yan, H.; Kouyi, G.L. Artificial neural network modelling in simulation of complex flow at open channel junctions based on large data sets. *Environ. Model. Softw.* **2014**, *62*, 178–187. [\[CrossRef\]](#)
9. Drikakis, D.; Sofos, F. Can artificial intelligence accelerate fluid mechanics research? *Fluids* **2023**, *8*, 212. [\[CrossRef\]](#)
10. Yang, H.C.; Chang, F.J. Modelling combined open channel flow by artificial neural networks. *Hydrol. Process. Int. J.* **2005**, *19*, 3747–3762. [\[CrossRef\]](#)
11. Yuhong, Z.; Wenxin, H. Application of artificial neural network to predict the friction factor of open channel flow. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 2373–2378. [\[CrossRef\]](#)
12. Fang, R.; Sondak, D.; Protopapas, P.; Succi, S. Neural network models for the anisotropic Reynolds stress tensor in turbulent channel flow. *J. Turbul.* **2020**, *21*, 525–543.
13. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [\[CrossRef\]](#)
14. Vinuesa, R.; Brunton, S.L. Enhancing computational fluid dynamics with machine learning. *Nat. Comput. Sci.* **2022**, *2*, 358–366. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Vinuesa, R. Perspectives on predicting and controlling turbulent flows through deep learning. *Phys. Fluids* **2024**, *36*, 031401. [\[CrossRef\]](#)
16. Matsubara, K.; Mitsuishi, A.; Iwamoto, K.; Murata, A. Prediction of pulsating turbulent pipe flow by deep learning with generalization capability. *Int. J. Heat Fluid Flow* **2023**, *104*, 109214. [\[CrossRef\]](#)
17. Xu, B.; Deng, J.; Liu, X.; Chang, A.; Chen, J.; Zhang, D. A review on optimal design of fluid machinery using machine learning techniques. *J. Mar. Sci. Eng.* **2023**, *11*, 941. [\[CrossRef\]](#)

18. Xu, Y.; Gan, X.; Pei, J.; Wang, W.; Chen, J.; Yuan, S. Applications of artificial intelligence and computational intelligence in hydraulic optimization of centrifugal pumps: A comprehensive review. *Eng. Appl. Comput. Fluid Mech.* **2025**, *19*, 2474675. [[CrossRef](#)]
19. Sarkar, M.; Samanta, A.; Sarkar, D.; Das, R.; Sarkar, S. Turbulence in a wall-wake flow downstream of two horizontal cylinders. *Mar. Georesour. Geotechnol.* **2023**, *42*, 878–897. [[CrossRef](#)]
20. Samanta, A.; Sarkar, M.; Mondal, H.; Das, R.; Sarkar, S. Turbulence anisotropy in a wall-wake flow downstream of two horizontal cylinders. *Flow Meas. Instrum.* **2023**, *94*, 102456. [[CrossRef](#)]
21. Samanta, A.; Mondal, H.; Sarkar, S. Turbulent bursting and higher-order moments in the wake flow behind two horizontal cylinders. *Acta Geophys.* **2025**, *73*, 4583–4604. [[CrossRef](#)]
22. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
23. Rehman, K.U.; Çolak, A.B.; Shatanawi, W. Artificial neural networking (ANN) model for convective heat transfer in thermally magnetized multiple flow regimes with temperature stratification effects. *Mathematics* **2022**, *10*, 2394. [[CrossRef](#)]
24. Samanta, A.; Mondal, H. Prediction model based on artificial neural network and bivariate spectral quasi-linearization method for compressible turbulent boundary-layer flow over a smooth flat surface. *Phys. Fluids* **2023**, *35*, 125148. [[CrossRef](#)]
25. Seawram, S.; Nimmanterdwong, P.; Sema, T.; Piemjaiswang, R.; Chalermssinsuwan, B. Specific heat capacity prediction of hybrid nanofluid using artificial neural network and its heat transfer application. *Energy Rep.* **2022**, *8*, 8–15. [[CrossRef](#)]
26. He, W.; Ruhani, B.; Toghraie, D.; Izadpanahi, N.; Esfahani, N.N.; Karimipour, A.; Afrand, M. Using of artificial neural networks (ANNs) to predict the thermal conductivity of zinc oxide–silver (50%–50%)/water hybrid Newtonian nanofluid. *Int. Commun. Heat Mass Transf.* **2020**, *116*, 104645. [[CrossRef](#)]
27. Kannaiyan, S.; Boobalan, C.; Nagarajan, F.C.; Sivaraman, S. Modeling of thermal conductivity and density of alumina/silica in water hybrid nanocolloid by the application of Artificial Neural Networks. *Chin. J. Chem. Eng.* **2019**, *27*, 726–736. [[CrossRef](#)]
28. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [[CrossRef](#)] [[PubMed](#)]
29. Müller, K.R.; Smola, A.J.; Rätsch, G.; Schölkopf, B.; Kohlmorgen, J.; Vapnik, V. Predicting time series with support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 999–1004.
30. Sahoo, B.B.; Jha, R.; Singh, A.; Kumar, D. Application of support vector regression for modeling low flow time series. *KSCE J. Civ. Eng.* **2019**, *23*, 923–934.
31. Cortes, C.; Vapnik, V. Support vector machine. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
32. Adun, H.; Wole-Osho, I.; Okonkwo, E.C.; Bamişile, O.; Dagbasi, M.; Abbasoglu, S. A neural network-based predictive model for the thermal conductivity of hybrid nanofluids. *Int. Commun. Heat Mass Transf.* **2020**, *119*, 104930. [[CrossRef](#)]
33. Parsaie, A.; Yonesi, H.A.; Najafian, S. Predictive modeling of discharge in compound open channel by support vector machine technique. *Model. Earth Syst. Environ.* **2015**, *1*, 1. [[CrossRef](#)]
34. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88. [[CrossRef](#)]
35. Ganaie, M.A.; Hu, M.; Malik, A.; Tanveer, M.; Suganthan, P. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.