# Application of Long-Short Term Memory and Convolutional Neural Networks for Real-Time Bridge Scour Prediction

Tahrima Hashem, Negin Yousefpour \*

Department of Infrastructure Engineering, The University of Melbourne, VIC, Australia

# Abstract

Scour around bridge piers is a critical challenge for infrastructures around the world. In the absence of analytical models and due to the complexity of the scour process, it is difficult for current empirical methods to achieve accurate predictions. In this paper, we exploit the power of deep learning algorithms to forecast the scour depth variations around bridge piers based on historical sensor monitoring data, including riverbed elevation, flow elevation, and flow velocity. We investigated the performance of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models for real-time scour forecasting using data collected from bridges in Alaska and Oregon from 2006 to 2021. The LSTM models achieved mean absolute error (MAE) ranging from 0.1m to 0.5m for predicting bed level variations a week in advance, showing a reasonable performance. The Fully Convolutional Network (FCN) variant of CNN outperformed other CNN configurations, showing a comparable performance to LSTMs with significantly lower computational costs. We explored various innovative random-search heuristics for hyperparameter tuning and model optimisation which resulted in reduced computational cost compared to grid-search method. The impact of different combinations of sensor features on scour prediction showed the significance of the historical time series of scour for predicting upcoming events. Overall, this study provides a greater understanding of the potential of Deep Learning algorithms for real-time scour prediction and early warning for bridges with distinct geology, geomorphology and flow characteristics.

*Key words:* Scour Prediction, Deep Learning, Time Series Forecasting, Hyperparameter Tuning, Long-short Term Memory Network, Convolutional Neural Network

## 1. Introduction

Bridge scour has been a major challenge for transportation systems around the world due to the significant damage and operational disruptions it can cause, and more impor-

<sup>\*</sup>Corresponding Author: negin.yousefpour@unimelb.edu.au Preprint submitted to journal

tantly the risk to public safety. In the US, for example, scour accounts for a significant number of bridge collapses, with more than 260,000 brides listed as vulnerable to scour failure (1). It is difficult to reliably predict the maximum scour depth for a bridge pier due to the complexity of the underlying causes of soil erosion in interaction with flow and structure. Uncertainties in riverbed material, flow, and geomorphological conditions, as well as climate change impacts, are the main challenges in the assessment of scour risk. For the last couple of decades, several research studies (2; 3; 4; 5) have focused on developing empirical models to estimate scour depth based on laboratory experiments and field observations. Most of these models often overestimate the depth of the scour (and in some cases underestimate) due to insufficient generalization to various conditions of the riverbed, flow, and structure (6).

Artificial Intelligence (AI) and Machine Learning (ML) for scour depth estimation have been investigated by a range of algorithmic strategies, such as evolutionary computing, fuzzy logic, artificial neural networks, support vector machines, and decision trees (see section 2). These proposed techniques have shown promising results in estimating maximum scour depth, often outperforming traditional empirical equations. However, the performance of these models is still limited by the availability and quality of training data, which are often scarce and may not cover a wide range of geological, hydraulic and geomorphological conditions. Furthermore, these models are typically trained to predict a maximum scour depth for a given flow discharge and cannot be used for dynamic scour prediction and real-time forecasting.

Yousefpour et al. (7; 8) pioneered the application of deep learning (DL) solutions, using Long Short Term Memory (LSTM) networks (9) for real-time scour forecasting. They leveraged the LSTM's superiority in temporal pattern recognition and capturing the underlying physics without direct feature extraction. They used historical scour monitoring data in Alaska to train DL models and predict future bed elevation variations and upcoming scour events. Fig. 1 shows the process of local scour around a bridge pier; scour happens when sediments are eroded and washed away by strong vortices in interaction with the bridge substructure, often accelerating during floods. Stage (flow/water level) and sonar (bed elevation) sensors are commonly used sensors installed on the bridge piers to monitor scour by collecting real-time flow and bed elevation data. The variants of LSTM models developed in (8) showed promising performance in forecasting the scour depth a week in advance, for case-study bridges in Alaska. However, there remain research questions about the capability of these algorithms to be upscaled for implementation in practice and deployment in different locations with variable meteorological, geological, geomorphological, and flow conditions.

In this study, we addressed fundamental research questions about the efficiency and generalization of the LSTM scour forecast models in diverse spatio-temporal domains. The LSTM models are compared against temporal CNN as an alternative deep learning solution, in terms of performance and computational cost. Random-Search (RS) heuristics for hyperparameter (HP) tuning are evaluated (11) to explore alternate com-



Fig. 1: An example of the bridge scour, adopted from (10).

putationally efficient methods to Grid-Search (GS) used in (8) for model optimisation. We also investigated the impacts of monitored features (input parameters), including stage, sonar, and velocity (discharge), on the performance of scour forecast models.

To summarise, four research questions were addressed in this study:

(RQ1) How do LSTM scour forecast models perform in predicting recent years of scour (Alaska) and how well do these models perform in a totally different location (Oregon)?

(RQ2) How do other competitor models, including CNN with vanilla and dilated convolutions, compare to LSTMs for the prediction of scour?

(RQ3) What computationally efficient heuristics can be adopted to find optimal model configurations to scour forecast DL models without sacrificing performance?

(RQ4) To what extent combinations of different sensor features can impact the performance of scour forecast DL models, and what are the most optimum feature combinations?

We carefully designed experiments and developed heuristics to investigate the aforementioned research questions as discussed in the following sections. The DL models were trained on the University of Melbourne's Spartan high-performance computing (GPU and CPU) cluster. The models were developed in Python using Tensorflow library (12).

# 2. Background

### 2.1. Maximum Scour Prediction

Since the 1940s, researchers have been using laboratory experiments and real-world observations to formulate empirical equations to estimate maximum scour depth. The

famous equations HEC-18 (10), FDOT (13), and CIRIA (14) equations were derived by multivariate analysis, considering various factors that impact the scour process. These equations have a large number of input and model parameters and measurement of the inputs is complex. However, often small databases (a few dozen to a few hundred samples) are employed for model calibration (15). Also, these empirical equations follow particular mathematical forms, such as linear, exponential, and hyperbolic terms, limiting their flexibility in capturing scour as a nonlinear phenomenon. As a result, the recommended equations often underperform in application to a broad range of scour conditions.

With the surge of data-driven methodologies, more sophisticated approaches from machine learning have been implemented for scour depth estimation, including Support Vector Machines (SVM) (16; 17; 18), Genetic Algorithms (19; 20), Decision Trees (21; 22), Artificial Neural Networks (ANNs) (23; 24), and CNNs (25). The results show that machine learning-based methods are usually more accurate than traditional equations in maximum scour depth prediction. However, these ML models face limitations when encountering unseen cases outside the convex hull of the training data, which can lead to poor generalization.

## 2.2. Real-time Scour Monitoring and Forecast

Although the worst-case scenario (usually a 1-in-100-year flood) is considered for estimating maximum scour depth in bridge design, many uncertainty factors could still lead to scour-related bridge failures: Larger flood events can occur due to extreme weather conditions and climate change; complexities in the scour process, particularly river sediment material interaction with flow which can result in larger scour depth; older bridges could fall behind the best recommended practices for scour estimation (26; 27). Considering all these factors, the transport authorities in the past two decades have been resorting to regular monitoring to manage scour, including regular inspections and surveys as well as advanced remote sensing.

In recent years, real-time scour monitoring sensors have become a prominent solution installed on many large-scale and critical bridges (28; 29; 30; 31). With the development of monitoring approaches, the sensors can facilitate data collection in real-time concerning bed elevation and other flow-related information. Compared to traditional databases (15), the sensor data collected is richer and more granular. Therefore, the monitoring data can also be utilized for scour depth prediction and providing early-warning information.

Yousefpour et al. (7; 8) pioneered the idea of incorporating real-time scour monitoring data to develop real-time scour forecast models using LSTMs. They incorporated monitoring data in Alaska (2006 to 2017) collected by the US Geological Survey (USGS) in partnership with the Departments of Transport (DOTs) in the United States. Specifically, historical time series of flow depth and bed elevation were used to train the LSTM models to predict future bed elevation and upcoming scour depth. The proposed model showed promising results in forecasting bed-elevation changes for at least seven days in advance. Compared to traditional empirical equations, these time series forecasting models yielded more reliable predictions of the upcoming scour trends and maximum depth of scour.

Chang et al. (32) presented a real-time monitoring system, using micro-cameras mounted on bridge piers, in conjunction with image processing and pattern recognition methodologies, to monitor variations in bed levels. This approach was validated through rigorous laboratory experiments, illustrating that the image data can be transformed to capture the progressive alterations in scour depth. Typically, monitoring sensors are installed on bridges deemed high risk.

To leverage limited data to better estimate scour risks across a bridge network, Maroni et al. (33) propose a scour hazard model tailored for road and railway bridges. By using a Bayesian network, the model decreases uncertainty in scour depth assessments at unmonitored bridges. This approach was demonstrated in a case study involving several Scottish road bridges.

Lin et al. (34) developed an early scour warning system installed at Da-Chia Bridge in Taiwan. They used R-CNN (a residual CNN) to read the real-time water level through CCTV images, and dedicated vibration-based arrayed sensors to obtain real-time local bed elevation of the target pier. These two data sources were incorporated to simulate upcoming hydrodynamic flows in fluvial rivers with hydraulic structures (e.g., bridge piers). The general scour is derived from a numerical model, and the local scour is estimated by empirical equations.

## 3. Scour Monitoring Data

We considered two scour monitoring databases for Alaska and Oregon provided by USGS to develop real-time scour forecast models:

- Alaska dataset: This includes previously collected sensor readings data from 2006 to 2017 from bridges over Sheridan River, Knik River, and Chilkat River with IDs 230, 539, and 742, respectively. Readers are referred to (8) for full data description. In addition, in this study we incorporated new data collected for 2018 to 2021 (for 742 bridge, 2019 and 2020 data are not available). An example of processed versus raw data for Alaska 539 bridge is provided in Fig. 2.
- Oregon dataset: This includes sensor readings from 2018 to 2021 for two bridges over the Trask and Luckiamute rivers (see Fig. 3).

For both datasets, each observation corresponds to an hourly recorded reading of the available sensors. We followed the same methodology as (7) to denoise, synchronize, and filter the raw monitoring data provided by USGS.



A strong negative correlation between stage and sonar can be observed for Alaska bridges. This is expected as higher flow velocity and depth result in increased scour depth, for most riverbed sedimentation. In addition, Alaskan bridges show consistent seasonal scour and filling patterns throughout a year, which is a key characteristic of Live-Bed scouring; each year, major scour followed by filling episodes are observed due to high flows (floods) happening around July/August and September/October. However, the scour patterns for the two bridges in Oregon are fundamentally different. Trask is a tidal river where bed material follows the tide cycles: scouring during outgoing tides and filling occurs during high tides. Also, the Luckiamute bridge shows a positive correlation trend between stage and sonar, contrary to expectations. Oregon USGS relates this to the presence of bed forms, which are structures such as sand dunes formed by the movement of bed material following the flow direction.

# 4. Approach

#### 4.1. Problem Definition

Given the historical sensor readings, i.e., water level: *Stage*, riverbed elevation: *Sonar*, and flow discharge: *Discharge* over time, we define the scour forecast task as a multivariate time-series prediction problem. Following is the mathematical definition of the scour forecast task.

**Definition 1.** Assume a series of past values  $\{x_{t-w_{in}+1}, x_{t-w_{in}+2}, \ldots, x_t\}$  for N sensors (features) with the historical window  $w_{in}$  where  $x_i \in \mathbb{R}^N$  and  $1 < w_{in} \le t$ . The scour forecasting task is to predict a series of future values,  $\{x_{t+1}, x_{t+2}, \ldots, x_{t+w_{out}}\}$  for the Sonar with the forecast window  $w_{out}$  where  $w_{out} \le t$ .



Fig. 3: Raw vs. Processed data - Oregon Luckiamute Bridge.

#### 4.2. Feature Engineering

In this study, we explored adding Discharge  $(X_{discharge})$  in addition to Stage  $(X_{stage})$  and Sonar  $(X_{sonar})$ , incorporated in our previous study (8) for Alaskan bridges <sup>1</sup>. Discharge represents the water volume passing the river cross section per unit of time.  $X_{discharge}$  which is obtained from the mean flow velocity measured by velocimeter sensors multiplied by the river channel cross-section area measured at the bridge location (35; 36).

We also engineered another feature, the equivalent velocity  $X_{eVelocity}$ , which is directly proportional to the flow velocity (V) following Equations. 1 to 4. By assuming a constant channel equivalent cross-sectional width  $\alpha$ , the channel cross section area can be estimated by  $\alpha \ge d$ , where d is the flow depth at a bridge pier equal to the difference between Sonar and Stage elevations (Equation 1).

$$d = X_{sonar} - X_{stage} \tag{1}$$

$$X_{discharge} = \alpha dV \tag{2}$$

$$X_{eVelocity} = \alpha V \tag{3}$$

$$X_{eVelocity} = X_{discharge} / (X_{sonar} - X_{stage})$$
<sup>(4)</sup>

The previous study showed that time features did not improve the performance of the LSTM models for Alaskan bridges. However, to enrich the periodicity information in Oregon's scour data, *time* features, namely *year-sin* and *year-cos* are incorporated in this study. These features are calculated by applying  $sin(\cdot)$  and  $cos(\cdot)$  transformations to the recorded timestep using Equation 5 and Equation 6, respectively.

<sup>&</sup>lt;sup>1</sup>for Oregon bridges, this sensor data was not available

$$year-sin = \sin(TS.2\pi/year)$$
(5)

$$year-cos = \cos(TS.2\pi/year)$$
(6)

The features *Sonar*, *Stage*, *Discharge* and *eVelocity* are denoted as sN, sT, dC and dV, respectively. Time features are denoted as y. The impact of various features and their combinations are discussed in Section 5.4.

# 4.3. Deep Learning For Scour Forecast

# 4.3.1. LSTM Models

Long Short Term Memory (9) is a specialised recurrent neural network (RNN) that is capable of modelling very large input and output sequences and does not suffer from the *gradient vanishing problem* - a typical problem in RNNs (37; 38). The principle analogy of LSTM is to remember the relevant short and long-standing information and discard the irrelevant information. Three different gates, i.e., *Forget* gate ( $\Gamma^f$ ), *Input* gate ( $\Gamma^i$ ) and *Output* gate ( $\Gamma^o$ ), enable the LSTM model to achieve this goal as shown in Fig. 4 (a). In these gates, the degree of relevance of information is quantified using a sigmoid ( $\sigma$ ) function with a value between 0 and 1. The larger the value, the higher the relevance. Equations (7 - 12) show how the hidden unit output ( $a_t$ ) and LSTM memory cell output ( $c_t$ ) are recurrently computed using these gates. *W* and *b* represent the weight and bias matrices.

$$\Gamma^f = \sigma(W_f[a_{t-1}, x_t] + b_f) \tag{7}$$

$$\Gamma^{i} = \sigma(W_{i}[a_{t-1}, x_{t}] + b_{i}) \tag{8}$$

$$\Gamma^o = \sigma(W_o[a_{t-1}, x_t] + b_o) \tag{9}$$

$$\tilde{c}_t = \tanh(W_c[a_{t-1}, x_t] + b_c)$$
 (10)

$$c_t = \Gamma^i \tilde{c}_t + \Gamma^f c_{t-1} \tag{11}$$

$$a_t = \Gamma^o \tanh(c_t) \tag{12}$$

Fig. 5 shows the architecture of the LSTM models for the multivariate time series forecasting of scour. The following three variants introduced in (8) are also implemented in this study:

- 1. Single-Shot (ss): In this model, one LSTM layer has been followed by one Dense and one Reshape layer. This model outputs the predictions for all the timesteps over the forecast window at once, as shown in Fig. 4(b).
- 2. Two-Layer, Single-Shot (ss2): Two LSTM layers are stacked together in this variant. The rest of the setting remains the same as Single-Shot.



(c) Fig. 4: (a) LSTM memory unit, (b) Single-Shot and (c) Feedback variants

3. Feedback (fb): In this variation, we hypothesize that the model's prediction accuracy would get improved, if we input the predicted value (as opposed to the actual value) of each timestep for the prediction of the next timestep. Therefore, in this variant, LSTM predicts one timestep ahead across the entire forecast window. Fig. 4 (c) illustrates the feedback process. The feedback variant also has one LSTM layer, similar to the Single-Shot model.

Model configuration nomenclature for an LSTM model is encoded as follows:

$$[model] - [w_{in}, w_{out}] - [units] - [dropout]$$

where, *model* is the LSTM variant,  $w_{in}$  and  $w_{out}$  are the length of model inputs and outputs, *units* is the number of LSTM units, and *dropout* denotes the dropout rate applied



Fig. 5: Multivariate time series modelling using LSTM model

to the model.

#### 4.3.2. CNN Models

CNN models proposed in (39) are widely used in computer vision for tasks like image classification (40) and object detection (41). They offer excellent performance with lower computational requirements. CNNs also excel in time series modelling tasks (42; 43; 44). We investigated three CNN variants for multi-variate time-series scour forecasting with an architecture as shown in Fig. 6).



Fig. 6: Multivariate time series modelling using CNN model

In time-series forecast, a pivotal element of the CNN model is the convolution process—a filter sliding over input time-sequence for feature extraction as depicted in Fig. 7. Equation 13 defines the standard convolution operation ( $C_{\text{vanilla}}$ ) as a dot product between a filter (F) and a time-sequence (S) of lengths k and l respectively. Because of the weight-sharing property of a CNN model, the entire time sequence is repeatedly convolved by the same filter. Fig. 7 (a) illustrates this vanilla convolution using a 3-step filter on a 9-step sequence with a stride of 1. Convolution results in a reduced-length sequence of 7 steps (l - k + 1 = 9 - 3 + 1 = 7), which leads to information loss. To prevent this loss, *padded* convolution ( $C_{padded}$ ) operates on an extended sequence S', created by padding zeros to the edges of S [see Equation 14 and Fig. 7 (b)]. This ensures the input and convolved time sequences have the same length.

$$C_{vanilla} := F.S_{t:t+k} \mid \forall t \in [1, l-k]$$
(13)

$$C_{padded} := F.S'_{t;t+k} \mid \forall t \in [0, l-k+1]$$
(14)

$$C_{casual} := F.S'_{t-k+1:t} \mid \forall t \in [1, l]$$
(15)

$$C_{dilated-casual} := F'.S'_{t-(k+d-1)+1:t} \mid \forall t \in [1, l]$$
(16)



Fig. 7: Illustration of different types of CNN convolutions for a 9-length time-series: (a) standard (vanilla) convolution process by a 3-length filter, (b) padded convolution, (c) casual convolution and (d) dilated-causal convolution by a 2-length filter.

Casual convolution ( $C_{casual}$ ) is often a preferable algorithm in time-series forecasting to avoid information leakage from the future (45). It precedes the input sequence with k-1 zeroes before convolution, as illustrated in Fig. 7 (c). Casual convolution would require a large filter or a deep network when it needs to examine a large number of past observations. Dilated Convolution overcomes this drawback by introducing a dilation factor d into the filter so that convolution can be applied on an input area larger than the original filter size, i.e., capturing larger contextual information. The modified filter F' is obtained by dilating the original filter with (d-1) zeros as shown in Fig. 7 (d). Dilation is performed using d = 2. The positions of zero correspond to the skipped timesteps in the input sequence. Both the casual (in Equation 15) and dilated casual convolutions  $(C_{dilated-casual})$  in Equation 16 output a sequence with the same length of the input. In the following, we introduce three CNN model configurations to perform the multivariate scour forecast:

1) Vanilla Convolutional Network (VCN): The convolution block of the VCN model comprises one layer of standard convolution (from Equation 13) and one layer of nonlinear activation function, i.e., Rectified Liner Unit (ReLU). Two identical convolution blocks are stacked together as a feature extractor that precedes a flattened layer, a dropout layer, and a fully connected dense layer. Finally, a reshape layer is added to ensure that the predicted time sequence contains the same number of sensor variables as input.

2) Fully Convolutional Network (FCN): A FCN ensures that the length of the convolved sequence is the same as the input. Here, the zero-padded convolution from Equation 14 with a stride value of 1 is used. We followed the structure adopted by Shelhamer et al. (46) for the FCN in this study, where the main block consists of one convolution layer, one batch normalization (BN) layer, and one ReLU layer. BN layer (47) ensures fast convergence and improves generalisation performance. The rest of the setup remains the same as the VCN.

3) Dilated Causal Network (DCN): As shown in Equation 16 and Fig. 7 (d), dilation factors of 1 and 2 are applied in two convolution layers. This enables the CNN model to perform causal convolution at a fine-grained level on the input sequence and a coarsegrained level on the convolved sequence (from the lower layer). Each convolution layer is followed by one BN layer and one ReLU layer. Model configuration nomenclatures for a temporal CNN are encoded as follows (with k and F being the filter size and no. of filters of the layers, respectively):

$$[model] - [k_1] - [F_1] - [k_2] - [F_2] - [dropout]$$

where the model is different CNN variants introduced,  $k_1$  and  $k_2$  are the kernel size,  $F_1$  and  $F_2$  are the hidden size of the dense layer, and dropout is the dropout rate applied to the model.

#### 4.4. Model Evaluation Criteria

We used Mean Square Error (MSE) on the *Sonar* readings to train the scour forecast model. For *S* no. of samples in a batch (32) with  $x_{i,t}$  and  $\hat{x}_{i,t}$  representing the actual and predicted values of *Sonar* feature  $X_{sonar}$  at timestep *t* for the *i*<sup>th</sup> sample, the training loss is computed as follows.

$$MSE = \frac{1}{S * w_{out}} \sum_{i=1}^{S} \sum_{t=1}^{w_{out}} (x_{i,t} - \hat{x}_{i,t})^2, x_{i,t} \in X_{sonar}$$
(17)

We evaluated the scour prediction performance of a model using the metric MAE on the *Sonar* readings. For *S* number of samples in a batch with  $x_{i,t}$  and  $\hat{x}_{i,t}$  representing the actual and predicted values of *Sonar* feature  $X_{sonar}$  at timestep *t* for the *i*<sup>th</sup> sample, the

performance metric, MAE is computed as follows.

$$MAE = \frac{1}{S * w_{out}} \sum_{i=1}^{S} \sum_{t=1}^{w_{out}} |x_{i,t} - \hat{x}_{i,t}|, x_{i,t} \in X_{sonar}$$
(18)

We used the hold-out method (48; 49) for evaluating the performance of the models. In the same unit as sensor readings, MAEs are reported in foot elevation throughout.

# 4.5. Hyperparameter Tuning

DL model optimisation and HP tuning using grid-search is computationally expensive (37). Following the grid-search approach in (8), the performance of all possible HP combinations was analysed by running each configuration for a large number of iterations. To explore a more computationally efficient alternative to this approach, we investigated a random-search strategy based on the following three possible policies to select the best model configurations. In this approach, a sample of HP combinations with 50%, 35%, and 67% sample size(s) out of all possible combinations are randomly selected in a number of trials (t = 20).

The grid-search results were used as a benchmark for the "true" performance of each configuration and to evaluate the outcome of these random search strategies.

- 1. *meanMAE Policy*: Model configurations are ranked based on the average MAE values across random trials. Lower average MAE values lead to higher rankings; this is irrespective of the number of times a model configuration appears across all trials (=f).
- medianMAE Policy: Model configurations are ranked by taking the median of the MAE values across the trials. Models with smaller median MAE would receive a higher rank than the model with larger median values; similar to mean MAE policy, this is irrespective of the number of times a configuration appears across all trials.
- 3. Bagging (Bootstrap Aggregation) Policy (50): In this policy, top k configurations with the lowest MAE values are selected on each trial. Considering  $f_{topk}$  as the number of times a particular configuration is ranked as top-k across t trials, k models (out of possible k \* t models) are finally selected based on  $f_{topk}$ ; model configurations with more frequent appearances in the top-k (i.e., larger  $f_{topk}$ ) receive higher ranks.

Finally, to observe how well the trained models generalize on unseen data, we generated prediction plots on the test datasets for the top-5 models. These top-5 models were selected after careful observations of the mean and standard deviation of MAE scores from both the random and grid-search policies. The prediction plots evaluate the performance of DL models in recognizing the scour (and filling) trends across time.

## 4.6. Experiments

Table 1 provides a summary of experiments we conducted to address the research questions outlined in Section 1. For all experiments except for Feature Combination Analysis, sN and sT are used as input features.

Table 1: Description of the experiments and their corresponding datasets, hyperparameters, and tunning values.

Experiment	Dataset	HPs	Tuning Values
1. HP	Alaska 742	$(w_{in}, w_{out})$	{(168, 168), (336, 168), (720, 168)}
Tunning Strategy		units	{32, 64, 128}
		models	{ss}
		dropouts	{0.0, 0.2}
2. LSTM	Alaska 230,539, 742	$(w_{in}, w_{out})$	{(168, 168), (336, 168), (720, 168)}
Evaluation		units	{32, 64, 128, 256}
	Oregon Trask, Luckiamute	models	$\{ss, ss2, fb\}$
		dropouts	{0, 0.2}
		$(w_{in}, w_{out})$	{(168, 168), (336, 168), (720, 168)}
	Alaska 230, 539, 742	layer-1 filter length: $k_1$	{3, 5, 7}
3. CNN	Oregon Trask, Luckiamute	layer-2 filter length: $k_2$	{5, 7, 11}
Evaluations		layer-1 filters: $F_1$	{64, 128, 256}
		layer-2 filters: $F_2$	{128, 512, 1024}
		models	{vcn, dcn, fcn}
		dropout	{0.0, 0.2}
4. Feature	Alaska 230, 539	features	{sN, sT, dV, dC, sNsT, sNdC, sNdV,
Combinations	Oregon Luckiamute		sTdV, sTdC, sNsTdV, sNsTdC}

# 5. Results

## 5.1. Evaluation of Random Policies for HP Tuning (Experiment 1)

We evaluated the HP selection policies on Alaska 742 bridge, using the single-shot LSTM scour forecast models. Given the past 1, 2, 3 weeks (168, 336, 720 hrs) observations ( $w_{in}$ ), we forecast hourly flow and bed elevation variation, aiming to predict scour for the upcoming week ( $w_{out}$ ). Here, the full search space comprises 18 distinct HP configurations as per Table 1. TALOS's *uniform mersenne* strategy (51) is used for random selection of HPs.

Fig. 8 demonstrates the results of HP tunning using grid-search and random-search policies for the 18 scour forecast models. Top-3 models with the smallest average MAE from grid-search (GS) served as benchmarks when evaluating random-search (RS) policies. Since the distribution of MAE values is not heavily skewed as observed in the boxplots, the mean values are used to rank the models in Fig. 8 (a). It is noted that higher ranked models (top-5) would remain the same with median criteria. Note that all the reported MAE values throughout the paper are in m (of elevation).

The meanMAE and medianMAE random-search policies show similar trends in lower sample sizes (35% and 50%), but as expected, they deviate by increasing the



Fig. 8: Comparison of random-search (RS) policies with grid-search (GS) for HP tuning: Ranked LSTM model configurations based on grid-search (a), showing all possible HP combinations for 18 models (b), frequency, top-k frequency, mean MAE and median MAE of HP configurations appearing in random-search iterations for (c) 35%, (d) 50%, (e) 67% sample size (top-3 models picked based on each policy are shown at top.

sample size. With a growing sample size, the mean-MAE estimations seem to converge towards grid-search estimates. The bagging policy also fails to find the top-ranked models when using small sampling sizes; however, with increasing sample size (67%), this method shows to be comparatively more robust and less sensitive to the poor-performing models compared to the meanMAE and medianMAE policies (e.g. models 12 to 18 show low ranking based on bagging but rank high in mean and median policies). Furthermore, the meanMAE and bagging policies seem to agree with 2 out of 3 top selected configurations, whereas the medianMAE policy fails to identify any of the top-performing models [Fig. 8 (e)].

## 5.2. LSTM Evaluation (Experiment 2)

The three variants of scour forecast LSTM models were evaluated to explore performance on 1) the recent years of collected sensor data from Alaska case study bridges, 230, 539, and 742 and 2) Oregon case study bridges, Trask and Luckiamute, with drastically different scour behaviour. Alaska and Oregon LSTM models were trained using the hold-out method, by partitioning the data into training, validation, and test subsets, with a splitting ratio (%) of 70-20-10 and 60-20-20, respectively.

Table 2: Top-5 LSTM models selected by random-search meanMAE policy for Alaska bridge 230, 539 and 742 and two Oregon bridges, Trask and Luckiamute.

Bridge	Rank	Models	meanMAE
230	1	ss-(336,168)-128-0	0.078
	2	ss2-(336,168)-32-0	0.083
	3	ss-(336,168)-32-0.2	0.083
	4	ss-(720,168)-64-0.2	0.083
	5	ss-(336,168)-32-0	0.083
	1	ss2-(720,168)-128-0	0.243
39	2	ss-(720,168)-64-0	0.253
	3	ss-(168,168)-128-0	0.259
<i>v</i> ,	4	ss2-(168,168)-64-0	0.266
	5	ss2-(720,168)-32-0	0.27
742	1	ss-(168,168)-128-0	0.41
	2	ss-(336,168)-64-0.2	0.413
	3	ss-(336,168)-128-0	0.419
	4	ss-(336,168)-32-0	0.424
	5	ss-(720,168)-32-0.2	0.431
	1	ss-(168,168)-32-0	1.573
~	2	ss-(168,168)-64-0.2	1.615
Trasl	3	ss-(168,168)-32-0.2	1.62
	4	ss-(168,168)-64-0	1.638
	5	ss-(336,168)-32-0.2	1.648
Luckiamute	1	ss-(720,168)-32-0	0.249
	2	ss-(720,168)-64-0	0.251
	3	ss-(720,168)-32-0.2	0.253
	4	ss-(720,168)-128-0	0.257
	5	ss2-(720,168)-32-0.2	0.261

The HP configurations (see Table 1) are optimized based on random-search method with meanMAE policy using %67 sample size. The top-5 LSTM model configurations for each bridge are presented in Table 2. The best-performing models achieve mean

MAE values ranging from 0.078 to 0.410 ft (0.12 m) for Alaska bridges and 0.249 to 1.573 ft (0.48m) for Oregon bridges.

The Single-Shot (ss) variant shows consistently better performance than Feedback (fb) variant for all bridges. Models with larger number of past observations, e.g.,  $w_{in} = 720$ , obtain smaller mean errors and appear as higher rank models. An input width 2 or 3 times the output/target width (forecast window) seems to yield the most accurate scour predictions.

#### 5.2.1. Scour Forecast for Alaska Bridges

Fig. 9 and Fig. 10 present the predicted values of *Sonar* (bed elevation) and *Stage* (water elevation) by the top-5 models for bridge 230 and 539 across the test dataset, between July-2021 to Sep-2021. The performance of models is assessed based on bed elevation (scour depth) predictions.

The models show reasonable prediction accuracy and are able to capture the scour and filling trends a week in advance throughout the test (unseen) period of the data.



Fig. 9: Stage (top) and Sonar (bottom) predictions of top-5 LSTM models selected by the random-search (meanMAE policy), mean and 95% confidence intervals over the test dataset - Alaska bridge 230.

For bridge 230, during the peak time of scour (August), the maximum error between the predicted and the actual value of scour is around 2ft (0.6m), whereas for the 539 bridge, it is around 1ft (0.3m). The inverse relation of the scour trend with the stage-water is well captured by the LSTM models with negotiable lags for all bridges.

#### 5.2.2. Scour Forecast for Oregon Bridges

Stage and Sonar prediction of top-5 models over the test dataset is presented in Fig. 11 and Fig. 12 for Luckiamute and Trask bridges, respectively. The test data is se-



Fig. 10: Stage (top) and Sonar (bottom) predictions of top-5 LSTM models selected by the random-search (bagging policy), mean and 95% confidence intervals over the test dataset - Alaska bridge 539.



Fig. 11: Stage (top) and Sonar (bottom) predictions of top-5 models selected by the meanMAE policy, showing the mean and 95% confidence interval over the test dataset - Luckiamute bridge.



Fig. 12: Stage (top) and Sonar (bottom) predictions of top-5 models selected by the meanMAE policy, showing mean and 95% confidence interval over the test dataset - Trask bridge.

lected for the period between April-2021 to July-2021 and August-2021 to Nov-2021, for Luckiamute and Trask, where significant scour and fillings are observed.

For Oregon Luckiamute, some of the large spikes in bed elevation seem to be due to errors in reading, which were not picked up by pre-processing techniques (verified with Oregon USGS). Yet, LSTM models are able to identify the realistic trend of bed elevation variation. For Trask which is a tidal river showing frequent daily fluctuations due to low and high tides no significant scour/filling is observed with less than 2ft (0.6m) change in bed level. The mean predictions fail to capture these frequent fluctuations, resulting in an underestimation of scour depth. However, the lower-bound values seem to capture maximum scour depths with some lag. Nevertheless, the Trask bed elevation variation is less than a meter and does not show significant scouring.

Overall, the LSTM models perform better in capturing peak values of scour and fill episodes for Alaska bridges compared to Oregon in our case studies. This can be attributed to the distinct scour process for the two Oregon bridges. Both Oregon bridges are over coastal rivers with soft slit sediments and bedforms (dunes). Sonar and Stage do not show the typical reverse correlation as in Alaskan bridge with riverine flows and consistent seasonal patterns (52; 10). In addition, the data availability in these locations is significantly less than in Alaska (3 years of historical data for Oregon versus 15 years for Alaska).

#### 5.2.3. Sequential Training on Oregon Data

The hold-out training method is sufficient for the LSTM models to learn the underlying patterns of scour/flow characteristics at coastal rivers such as Trask and Luckimaute, leading to poor generalization. Cross-validation (48; 49) could be an alternative, however it is computationally expensive. Therefore, motivated by transfer learning (53), we implemented a "Sequential Training" technique as shown in Fig. 13 to improve the forecast accuracy for Oregon bridges. For this experiment, we used the top-3 LSTM model configurations from Section 5.2.2.



Fig. 13: Illustration of 3-fold sequential training and validation.

The major part (90%) of the data is repeatedly split into 3 Folds and 5 Folds, with a small portion (10%) set aside for the final evaluation. Each of the folds is further divided into three parts: train, validation, and test. The model is trained using the folds in a sequential manner, i.e., the model trained (with its learned weights and biases) in a fold is going to be fine-tuned using the newer train data from the next fold. There is no overlap across the training data between two adjacent folds. This data division and the transfer of "learning" between folds saves significantly on training time and expected to improve the performance of DL models.

Table 3 presents the MAE values on Sonar, and Fig. 14 shows the time-series forecast over the final-test data using the 3-Fold and 5-Fold sequentially-trained LSTM models for Trask bridge. It is observed that the prediction error is reduced using the Sequential Training method.



Fig. 14: Bed elevation (Sonar) predictions using top LSTM models with 5-Fold and 3-Fold sequential training method for Oregon Trask bridge.

		model-1		model-2		model-3				
		ss-(168,168)-32-0		ss-(168,168)-64-0.2			ss-(168,168)-32-0.2			
SQ	fold	val	test	final_test	val	test	final_test	val	test	final_test
Hold-out	N/A	1.57	N/A	0.66	1.62	N/A	0.72	1.62	N/A	0.71
3-Fold	0	0.79	0.91	0.88	0.87	1.03	0.89	0.67	0.95	0.76
	1	0.45	0.74	0.59	0.47	0.82	0.53	0.51	0.81	0.56
	2	2.35	0.36	0.61	2.21	0.38	0.57	2.49	0.31	0.62
5-Fold	0	0.42	0.63	0.66	0.42	0.69	0.79	0.37	0.67	0.84
	1	0.95	0.62	0.66	1.01	0.69	0.76	1.17	0.60	0.71
	2	0.38	0.35	0.53	0.60	0.34	0.60	0.64	0.35	0.62
	3	1.15	2.81	0.64	1.21	3.17	0.59	1.25	2.90	0.64
	4	0.36	0.14	0.40	0.60	0.21	0.42	0.50	0.14	0.38

Table 3: MAE of 3-Fold and 5-Fold sequentially-trained LSTM models - Trask bridge.

# 5.3. CNN Evaluation (Experiment 3)

Three different variations of CNN models, i.e., VCN, DCN, and FCN as described in Section 4.3.2 are trained for bridge 230 and 539 in Alaska and Luckiamute bridge in Oregon. *Stage* and *Sonar* are considered as both the input and target features in these models. Table 1 provides the tuning range for the relevant hyperparameters. Gridsearch is used to tune the hyperparameters of CNN models. The splitting ratios (trainvalidation-test) are considered as 60%-30%-10% and 60%-20%-20% for the Alaska and Oregon bridges.

Bridge	Vanilla Convolutional	avg.	Dilated Causal	avg.	Fully Convolutional	avg.
	Netwrok (VCN)	MAE	Network (DCN)	MAE	Network (FCN)	MAE
230	vcn-5-64-9-128-0	0.090	dcn-9-64-9-128-0	0.107	fcn-5-64-7-64-0	0.086
	vcn-7-64-9-128-0	0.091	dcn-5-64-5-128-0	0.107	fcn-3-64-5-128-0	0.086
	vcn-7-128-9-256-0	0.092	dcn-7-64-7-128-0	0.108	fcn-9-64-7-128-0	0.086
	vcn-7-128-9-256-0.2	0.093	dcn-3-64-3-64-0	0.109 fcn-3-128-5-128-0		0.087
	vcn-7-64-9-128-0	0.093	dcn-3-128-5-64-0	0.109	fcn-5-64-7-128-0	0.088
539	vcn-9-128-9-64-0.2	0.382	dcn-5-128-5-64-0	0.323	fcn-3-256-5-128-0	0.315
	vcn-9-128-7-64-0	0.384	dcn-5-64-3-64-0	0.327	fcn-7-256-7-128-0.2	0.317
	vcn-9-64-9-64-0.2	0.389	dcn-7-128-5-64-0.2	0.328	fcn-5-128-7-256-0	0.318
	vcn-9-64-9-64-0	0.389	dcn-5-256-7-256-0	0.329	fcn-9-64-7-128-0	0.319
	vcn-9-64-7-64-0	0.391	dcn-7-64-7-128-0	0.330	fcn-9-128-7-64-0.2	0.319
Luckiamute	vcn-3-64-3-64-0.2	0.525	dcn-5-256-3-256-0.2	0.305	fcn-5-128-5-256-0.0	0.278
	vcn-5-128-5-64-0.0	0.549	dcn-3-128-5-256-0.2	0.308	fcn-5-256-5-256-0.0	0.282
	vcn-5-64-5-64-0.2	0.583	dcn-5-256-5-256-0.2	0.309	fcn-5-256-5-256-0.2	0.283
	vcn-5-64-7-128-0.2	0.606	dcn-3-256-3-256-0.2	0.309	fcn-3-128-5-256-0.0	0.285
	vcn-5-64-7-64-0.2	0.610	dcn-5-128-5-256-0.0	0.310	fcn-5-256-3-256-0.0	0.286

Table 4 summarizes the performance of the CNN models for scour prediction (Sonar). The FCN variant yields the best performances, as evidenced by the lower MAE scores in Table 4, in comparison with the VCN and DCN models for both Alaska and Oregon bridges. The best-performing CNN models achieve competitive accuracy for both

Alaska and Oregon bridges, compared with LSTM.



Fig. 15: Stage (top) and Sonar (bottom) predictions of top-5 FCN models selected by the meanMAE policy, mean and 95% confidence intervals over the test dataset - Alaska 230 bridge.



Fig. 16: Stage (top) and Sonar (bottom) predictions of top-5 FCN models selected by the meanMAE policy, mean and 95% confidence intervals over the test dataset - Alaska 539 bridge.

#### 5.3.1. Scour Forecast for Alaska Bridges

Fig. 15 and Fig. 16 demonstrate the time-series prediction of Sonar and Stage by the top-5 FCN models across the test dataset. The CNN models demonstrate a reasonable performance, comparable with LSTM models, capturing the trend of scour and filing a week in advance.

The CNN model with full convolution from Equation 14 shows that it outperforms other convolutional strategies in generalization and captures both Sonar and Stage trends. Particularly, DCN models fail to capture the scour and filling trends as compared to the VCN and FCN. This implies that skipping the intermediate observations by the dilated convolutions from Equation 16 did not result in higher accuracy. One reason behind the superior performance by FCN can be the inclusion of the edging values while making batch-to-batch transitions via the 0-*padded* convolution. This ensures no loss of information while learning the underlying scour patterns.

#### 5.3.2. Scour Forecast for Oregon Results

Fig. 17 illustrates the performance of top-5 FCN models over the test dataset, between April 2021 and July 2021. Similar to the LSTM models, the CNN models perform less accurately for Oregon data compared with Alaska, particularly in capturing the scour and fill trends. The error of mean Sonar prediction varies between 1.5 to 2.5ft (0.5 to 0.75m).



Fig. 17: Stage (top) and Sonar (bottom) predictions of top-5 FCN models. The mean and 95% confidence intervals are considered on the test dataset - Oregon Luckiamute bridge.

#### 5.4. Feature Combination Analysis (Experiment 4)

Different combinations of features are considered as input to LSTM models to evaluate their impact in scour prediction (wrapper method). The best Single-Shot LSTM configurations were trained for this purpose over two Alaska and Oregon bridges as shown in Figs. 18 and 19. Note that for Oregon bridges, the discharge data was not; therefore, dC dV are not included in the impact analysis.

The first term in model configuration nomenclatures indicates the feature combination, e.g. *sNsTdV*, indicates {*Sonar*, *Stage*, *Discharge*} input feature combination (refer to Section 4.2 for features notations).



Fig. 18: Feature impact evaluation - Alaska. The first term in model configurations indicates the feature combination.

These plots show that the input feature sets containing sN (Sonar) consistently outperform the feature sets without sN for all bridges. The models with sN as the only input feature shows the best performance for all bridges. The groups sNdV and sNdV show the second smallest error for all bridges, showing comparable performance as sN alone. As expected, Discharge and eVelocity have an equivalent impact on prediction performance as sNdC and sNdV models show consistently similar error ranges. Also, Stage seems to have a similar impact on prediction as Discharge and eVelocity, as observed in very close error ranges among sNsT and sNdC, and sNdV. Therefore, adding velocity features to DL models in Alaska does not seem to improve significantly scour forecasting as initially expected. This further fortifies the results reported in (8) about stage



Fig. 19: Feature impact evaluation - Oregon. The first term in model configurations indicates the feature combination.

time series having a high correlation with discharge time series, therefore embodying velocity characteristics required for LSTMs to learn the scour patterns.

For the Oregon bridges, the scour models do not obtain higher accuracy by including the time features denoted as y, similar to Alaska case studies in (8). Input feature sets: sN and sNsT obtain similar accuracy in predicting scour. Similar to Alaska bridges, LSTM models show poor performance when the input feature set does not contain the Sonar values.

The strong dependency among the recent and past values of the sonar readings is evidenced by the lower MAE of the scour model with Sonar as the only feature and the high autocorrelation coefficient of sonar readings. However, Stage is shown to be more impactful for scour prediction in the case of the Luckimaute bridge. This can be related to a lack of consistent periodic scour and filing patterns as opposed to Alaska bridges and Trask bridges with tidal river bed fluctuations. Nevertheless, it is evident from this experiment that flow-related times series on their own, i.e., stage, discharge and velocity, are inadequate for reliable scour prediction.

# 6. Conclusions

This paper investigated prominent Deep Learning (DL) algorithms for scour forecasting based on historic sensor monitoring data, including three variants of LSTM and CNN, for case-study bridges in Alaska and Oregon. The primary focus was to address the challenges of DL models for real-time scour prediction, including 1) performance optimization of algorithms and hyperparameter tuning, 2) generalisation to diverse locations with distinct geological/geomorphological and hydraulic/flow characteristics, and 3) impact of various combinations of sensor features relevant to scour.

The top-performing LSTM models achieved mean absolute error (MAE) ranging from 0.1m (in Alaska) to 0.5m (in Oregon). The best CNN model, the FCN variant, showed similar error ranges (0.1m to 0.75m). Both DL algorithms showed promising performance in forecasting scour for bridges in Alaska, with CNN models, particularly the FCN variant, achieving slightly lower prediction errors. However, CNN was trained with significantly lower computational costs running on the same GPU clusters. Higher errors were observed for Oregon bridges compared to Alaska, especially in capturing peaks of scour and filling episodes. This was related to the fundamentally different scour process in Oregon bridges, such as tidal flows and the presence of coastal bed forms, and the lack of substantial historical data. In addition, Oregon bridges did not show consistent periodical and seasonal flow and scour patterns as observed in Alaskan bridges.

To improve the accuracy of DL models for Oregon, we implemented a sequential training approach integrated with transfer learning. This training approach proved effective in improving the performance, while reducing the computational time.

For LSTM models, the single-shot model consistently outperformed the two-layer single-shot and feedback variants. Regarding the CNN, the FCN achieved lower MAE values on all bridges compared to the Vanilla Convolutional Network and Dilated Causal Network. Additionally, our feature impact analysis using the wrapper method showed adding Discharge, and Velocity time series to Stage and Sonar did not improve the performance of DL models in Alaska. An interesting observation consistent across all bridges in Oregon and Alaska was that the models without Sonar (bed elevation) time series showed significantly higher errors. In fact, models with Sonar as the only input feature obtained similar (and in some cases slightly better) scour prediction accuracy compared with benchmark models with the Sonar plus Stage feature set. For Oregon bridges, the time features did not improve the performance of DL models similar to what was observed previously for Alaska.

The random-search method with mean MAE and Bagging heuristics introduced in this study proved to be efficient alternatives to grid-search for hyperparameter tunning. The top LSTM configurations identified by the random-search with higher sample sizes (%67) resulted in finding the most optimal configurations, while saving significantly on computational cost.

In conclusion, this study highlights the potential of deep learning models, particularly LSTM and CNN variants, for accurate scour forecasting in diverse bridge locations. However, challenges remain in adapting these models to fundamentally different scour processes and limited historical data, as observed in the Oregon bridges. Future research should focus on developing more robust and adaptable models that can handle varying scour characteristics and data limitations. Additionally, integrating domain knowledge and physics-based modelling approaches with data-driven techniques could further improve the interpretability and generalization of scour prediction models. Finally, the exploration of advanced hyperparameter optimization techniques and transfer learning strategies may help reduce computational costs and improve the efficiency of model development for scour forecasting applications.

## 7. Acknowledgement

The funding for this research was provided by The University of Melbourne and Arup Global Research. We would like to express our gratitude to Greg Lind from USGS and Kira Glover-Cutter from Oregon DOT for providing the data and insights throughout this research. We also extend our thanks to Dr Bo Wang for his help in preparing this manuscript.

### References

- N. Yousefpour, Z. Medina-Cetina, J.-L. Briaud, Evaluation of unknown foundations of bridges subjected to scour: Physically driven artificial neural network approach, Transportation Research Record 2433 (1) (2014) 27–38. doi:10.3141/2433-04.
  - URL ttps://doi.org/10.3141/2433-04
- [2] D. M. Sheppard, B. Melville, H. Demir, Evaluation of Existing Equations for Local Scour at Bridge Piers, Journal of Hydraulic Engineering 140 (1) (2014) 14–23.
- [3] D. S. Mueller, C. R. Wagner, Field observations and evaluations of streambed scour at bridges, Tech. rep., United States. Federal Highway Administration. Office of Research. (2005).
- [4] B. Liang, S. Du, X. Pan, L. Zhang, Local scour for vertical piles in steady currents: Review of mechanisms, influencing factors and empirical equations, Journal of Marine Science and Engineering 8 (1) (2019) 4.
- [5] A. Pizarro, S. Manfreda, E. Tubaldi, The science behind scour at bridge foundations: A review, Water 12 (2) (2020) 374.
- [6] A. Sharafati, M. Haghbin, D. Motta, Z. Yaseen, The application of soft computing models and empirical formulations for hydraulic structure scouring depth simulation: A comprehensive review, assessment and possible future research direction, Archives of Computational Methods in Engineering 28 (11 2019).
- [7] N. Yousefpour, S. Downie, S. Walker, N. Perkins, H. Dikanski, Machine learning solutions for bridge scour forecast based on monitoring data, Transportation Research Record 2675 (10) (2021) 745–763.
- [8] N. Yousefpour, O. Correa, Towards an ai-based early warning system for bridge scour, Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards (2023) 1–27.
- [9] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
- [10] L. Arneson, Evaluating scour at bridges, Tech. rep., United States. Federal Highway Administration (2013).
- [11] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, Journal of Machine Learning Research 13 (2012) 281–305.
- [12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens,

B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015). URL http://tensorflow.org/

- [13] F. Office, Bridge scour manual, Tech. rep., Florida Department of Transportation (2021).
- [14] A. Kirby, M. Roca, A. Kitchen, M. Escarameia, O. Chesterton, Manual on scour at bridges and other hydraulic structures, Ciria, 2015.
- [15] D. M. Sheppard, H. Demir, B. W. Melville, Scour at wide piers and long skewed piers, Vol. 682, Transportation Research Board, 2011.
- [16] A. Goel, M. Pal, Application of support vector machines in scour prediction on grade-control structures, Engineering Applications of Artificial Intelligence 22 (2) (2009) 216–223.
- [17] H. Sharafi, I. Ebtehaj, H. Bonakdari, A. H. Zaji, Design of a support vector machine with different kernel functions to predict scour depth around bridge piers, Natural Hazards 84 (2016) 2145–2162.
- [18] M. Najafzadeh, A. Etemad-Shahidi, S. Y. Lim, Scour prediction in long contractions using anfis and svm, Ocean Engineering 111 (2016) 128–135.
- [19] H. M. Azamathulla, A. A. Ghani, N. A. Zakaria, A. Guven, Genetic programming to predict bridge pier scour, Journal of Hydraulic Engineering 136 (3) (2010) 165–169.
- M. Khan, H. M. Azamathulla, M. Tufail, A. Ab Ghani, Bridge pier scour prediction by gene expression programming, Proceedings of the Institution of Civil Engineers Water Management 165 (9) (2012) 481–493. doi:10.1680/wama.11.00008. URL https://doi.org/10.1680/wama.11.00008
- [21] M. Pal, N. Singh, N. Tiwari, Pier scour modelling using random forest regression, ISH Journal of Hydraulic Engineering 19 (2) (2013) 69–75.
- [22] D. Tien Bui, A. Shirzadi, A. Amini, H. Shahabi, N. Al-Ansari, S. Hamidi, S. K. Singh, B. Thai Pham, B. B. Ahmad, P. T. Ghazvinei, A hybrid intelligence approach to enhance the prediction accuracy of local scour depth at complex bridge piers, Sustainability 12 (3) (2020) 1063.
- [23] S.-U. Choi, S. Cheong, Prediction of local scour around bridge piers using artificial neural networks 1, JAWRA Journal of the American Water Resources Association 42 (2) (2006) 487–494.
- [24] T. Lee, D. Jeng, G. Zhang, J. Hong, Neural network modeling for estimation of scour depth around bridge piers, Journal of hydrodynamics 19 (3) (2007) 378–386.
- [25] J. Zhang, H. Zhao, A prediction model for local scour depth around piers based on cnn, in: 2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS), IEEE, 2020, pp. 318–320.
- [26] O. Khandel, M. Soliman, Integrated framework for quantifying the effect of climate change on the risk of bridge failure due to floods and flood-induced scour, Journal of bridge engineering 24 (9) (2019) 04019090.
- [27] R. Lamb, W. Aspinall, H. Odbert, T. Wagener, Vulnerability of bridges to scour: insights from an international expert elicitation workshop, Natural Hazards and Earth System Sciences 17 (8) (2017) 1393–1409.
- [28] P. F. Lagasse, Instrumentation for measuring scour at bridge piers and abutments, Transportation Research Board, 1997.
- [29] J.-L. Briaud, S. Hurlebaus, K.-A. Chang, C. Yao, H. Sharma, O.-Y. Yu, C. Darby, B. E. Hunt, G. R. Price, Realtime monitoring of bridge scour using remote monitoring technology, Tech. rep., Texas Transportation Institute (2011).
- [30] M. F. K. Lebbe, W. Lokuge, S. Setunge, K. Zhang, Failure mechanisms of bridge infrastructure in an extreme flood event, in: Proceedings of the 1st International Conference on Infrastructure Failures and Consequences (ICFC 2014), University of Southern Queensland, 2014.
- [31] L. Prendergast, K. Gavin, A review of bridge scour monitoring techniques, Journal of Rock Mechanics and Geotechnical Engineering 6 (2) (2014) 138–149. doi:https://doi.org/10.1016/j.jrmge. 2014.01.007.

URL https://www.sciencedirect.com/science/article/pii/S167477551400016X

- [32] W.-Y. Chang, L. Jihn-Sung, Y. Teng-Yi, L. Franco, L. Lung-Cheng, T. Whey-Fone, L. Chin-Hsiung, Pier scour monitoring system by bed-level image tracking, International Journal of Sediment Research 29 (2) (2014) 269–277.
- [33] A. Maroni, E. Tubaldi, D. V. Val, H. McDonald, D. Zonta, Using bayesian networks for the assessment of underwater scour for road and railway bridges, Structural Health Monitoring 20 (5) (2021) 2446– 2460.
- [34] Y.-B. Lin, F.-Z. Lee, K.-C. Chang, J.-S. Lai, S.-W. Lo, J.-H. Wu, T.-K. Lin, The artificial intelligence of things sensing system of real-time bridge scour monitoring for early warning during floods, Sensors 21 (14) (2021).
- [35] T. J. Buchanan, W. P. Somers, Discharge measurements at gaging stations, Techniques of waterresources investigations (1969).
- [36] D. P. Turnipseed, V. B. Sauer, Discharge measurements at gaging stations, Tech. rep., US Geological Survey (2010).
- [37] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, Neural Computation 12 (10) (2000) 2451–2471.
- [38] F. A. Gers, J. Schmidhuber, Recurrent nets that time and count, Proceedings of the International Joint Conference on Neural Networks 3 (June) (2000) 189–194.
- [39] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278 – 2324.
- [40] T. F. Gonzalez, Handbook of approximation algorithms and metaheuristics, Handbook of Approximation Algorithms and Metaheuristics (2007) 1–1432.
- [41] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).
- [42] J. C. B. Gamboa, Deep learning for time-series analysis, arXiv preprint arXiv:1701.01887 (2017).
- [43] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: A review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963.
- [44] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, ArXiv abs/1803.01271 (2018).
- [45] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).
- [46] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4) (2017) 640–651.
- [47] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, pmlr, 2015, pp. 448–456.
- [48] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: International joint conference on artificial intelligence, Vol. 14, Montreal, Canada, 1995, pp. 1137–1145.
- [49] A. K. Jain, R. C. Dubes, C.-C. Chen, Bootstrap techniques for error estimation, IEEE transactions on pattern analysis and machine intelligence (1987) 628–633.
- [50] E. Bauer, P. Chan, S. Stolfo, D. Wolpert, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, in: Machine Learning, 1999, pp. 105–139.
- [51] M. Matsumoto, T. Nishimura, Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, ACM Transactions on Modeling and Computer Simulation 8 (1) (1998) 3–30.
- [52] D. M. Sheppard, Bridge scour in tidal waters (1993).
- [53] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.