IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 17, 2022, accepted May 13, 2022, date of publication May 23, 2022, date of current version May 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3177201

Cost Effective Soft Sensing for Wastewater Treatment Facilities

MAIRA ALVI[1], TIM FRENCH[1], RACHEL CARDELL-OLIVER[1], PHILIP KEYMER[2], AND ANDREW WARD[2,3]

[1]Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA 6009, Australia
[2]Australian Centre for Water and Environmental Biotechnology, The University of Queensland, Brisbane, QLD 4072, Australia
[3]Urban Utilities Queensland, Brisbane, QLD 4006, Australia

Corresponding author: Maira Alvi (maira.alvi@research.uwa.edu.au)

This work was supported in part by the Cooperative Research Centre Projects (CRC-P) "Transforming wastewater treatment in regional Australia with robust technology for multiple benefits" through the Australian Government, Urban Utilities Queensland, under Grant CRCPSIX000079; and in part by the Ph.D. Scholarship from The University of Western Australia.

ABSTRACT Wastewater treatment plants are complex, non-linear, engineered systems of physical, biological and chemical processes operating at different timescales. Sensor systems are used to monitor wastewater treatment plants in order to ensure public safety and for efficient management of the plants. However, parameters of interest for wastewater can require expensive or inaccurate sensors or may require off-site laboratory analysis. For example, ammonium is important as a prime indicator of treatment efficiency and is highly regulated in discharge water. But ammonium sensors are also expensive at over $10,000 (AUD) per sensor. Soft sensors are computational models that accurately estimate process variables using the measurements from few physical sensors and can offer a cost-effective substitute for expensive wastewater sensors such as ammonium. In this paper, we propose a hybrid neural network architecture for learning soft sensors for complex phenomena. Our network architecture fuses sequential modelling with Gated Recurrent Neural Network units (GRUs) to capture global trends, with Convolution Neural Network (CNN) kernels to facilitate learning of local behaviours. We demonstrate the effectiveness of our technique using real-world data from a wastewater treatment plant with two-stage high-rate anaerobic and high-rate algal treatments. Secondly, we propose a novel data preparation algorithm that enables the deep learning techniques to learn from a limited data and facilitates fair evaluation. We develop and learn a soft sensor to predict ammonium and study its generalization. Our results demonstrate fit for purpose accuracy and that the soft sensor model is able to capture complex temporal patterns of the ground truth sensor time series. Finally, we publicly release an annotated data set of a secondary wastewater treatment plant to accelerate the research in the development of soft sensors.

INDEX TERMS Wastewater treatment, high rate algal ponds, ammonium, soft sensors, hybrid model, recurrent neural network, deep learning.

I. INTRODUCTION

Wastewater treatment plants (WWTPs) like many industrial processes are cyber-physical systems (CPSs) which are comprised of interconnected internet of things (IoT) devices, such as sensors and actuators, typically referred to as the inputs/outputs (I/O) of a programmable logic controller (PLC). For optimal treatment performance, several process variables are actively monitored via sensors to make informed decisions, be it automatically via integrated process control or via manual operator intervention. Automatically controlled process variables typically include easily measured parameters related to maintaining an operational condition within process (e.g. dissolved oxygen level). Whereas, licensed discharge water quality parameters, are not simple to quantify and monitor automatically [1]. These licensed water quality parameters often include the concentrations of ammonium, nitrates, total nitrogen, phosphates, and biological or chemical oxygen demand [2].

Unlike most industrial processes, wastewater treatment has minimal control over the feed to the system, hence expeditious quantification of process state indicators of

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang.

interest is important to allow for timely response to changing conditions. Broadly, the process state indicators can be partitioned into difficult to measure (primary) variables and easy to measure (secondary) variables. Here the level of difficulty is synonymous to the economic limitations, for instance the instrumentation is expensive, or availability of results is subject to time-delayed responses (e.g, offline laboratory analysis) [2]. In contrast to the difficult to measure variables, the secondary variables are easily captured via range of affordable and reliable instrumentation. Due to the nature of the treatment process, primary variables are contingent upon a range of other variables governed by the interrelated biochemical reactions occurring within the system. Under this lens, the easy to measure variables can be seen as weak surrogates of primary process variables. This naturally motivates us to explore and develop an algorithm that can closely approximate a primary variable by leveraging a range of commonly available secondary variables. Thus effectively, developing a soft sensor i.e. a computer program that can potentially replace the expensive instruments by easy to tune, low latency and a low cost solution.

In essence, soft sensors are computational models that aim to accurately estimate process variables that can either not be measured directly or require expensive instrumentation [3], [4]. These sensors are often data driven techniques that includes statistical and machine learning models and requires large volume of annotated data set. The existing approaches in literature include auto-regressive integrated moving average (ARIMA) [5], logistic regression [6], principal component analysis (PCA) [7], partial least square (PLS) [8], hidden markov model (HMM) [9], support vector machines (SVMs) [10], random forest (RF) [11], and artificial neural networks (ANNs) [12], [13]. In comparison, ANN models have demonstrated superior and robust performance by large margins.

ANN models are a specialized subset of machine learning algorithms that include hierarchical arrangement of numerous universal function approximators called neurons. The hierarchical arrangement also inspires its popular alias of "deep model". Several arrangements i.e., architectures, have been devised by the wastewater research community for inferential purposes of several process variables. However, most of the available works have discarded the temporal notion of the processes in inference model design [14]–[19], since the wastewater treatment processes inherently include a temporal lag in the process (e.g. collection via a distributed sewer network prior to centralised treatment at a WWTP with treatment processes lasting for days). We explore this further, and conjecture that explicit inclusion of time notion by designing a specialized design of data-driven pipeline can significantly improve the performance of soft-sensors.

The wastewater research community has explored and developed a range of soft sensing algorithms using machine learning and statistical learning. Among these techniques, variants of recurrent and convolutional neural networks are popularly used to model wastewater process parameters [20]–[23]. In contrast, the proposed architecture extracts temporal features from an input time series using GRUs and afterwards iteratively refines them with the aid of convolutional layers to perform regression. We argue that local variations in the learnt temporal features improves prediction performance. The GRUs layers encapsulate sequential information while considering the correlations among different input time series, and then CNN is applied for learning local correlations on the temporally enriched extracted features. The intuition behind this hybrid architecture is motivated by the fact that wastewater treatment processes are highly non-linear and the associated multivariate time series have convoluted correlations. Therefore, we need to extract distinctive features of each of the correlated multivariate time series. This is achieved by the application of CNNs filters on the GRUs output. We demonstrate that the combination of GRU and CNN unveils additional information that aids accurate and robust predictions.

The main contribution of this paper is the design of a novel soft sensor framework that quantifies ammonium concentration in real time. The soft sensor framework leverages ensemble of deep neural network architectures for accurate predictive modelling, and to enable cost-effective monitoring for these facilities. It is a novel hybrid model in the wastewater application domain that combines the strength of Gated Recurrent Neural Network (GRU) [24], and Convolution Neural Network (CNN) [25], we call it GRUconv. In contrast with existing techniques, it takes the temporal notion into account by modeling the process via recurrent units and further improves upon the features based on the local trends captured via convolutional layers. We illustrate the superior performance of our technique by a detailed study over ammonium - a critical and expensive indicator in wastewater treatment. Our exploration includes detailed empirical and qualitative comparison against existing available techniques in the wastewater literature. This includes CNN, GRU [24], LSTM [26], Bi-LSTM and Bi-GRU.

Our framework encapsulates the latent temporal variations in the input time series by hierarchical arrangement of GRU units. These units are recursive in nature i.e., process sequential data to learn the salient macro-temporal features. These temporal features are additionally refined based on the local information by repeated traversal through convolutional layers. Thus the model learns an embedding that can be linearly weighted to estimate a process variable. We empirically study the performance of our technique over the data gathered from a two stage high-rate anaerobic and high-rate algal treatment process as shown in Figure 1. The curated data set is composed of $\approx$ 12 months duration. The shorter duration makes it vulnerable to seasonality patterns and that yields imbalanced distribution in the data. In addition, it is also affected from sensors internal variance, meteorological variation, missing data and other external perturbations. These practical issues makes training of soft-sensors a challenging task.

Deep models are known to require huge amounts of data for training. In presence of restricted amount of data and

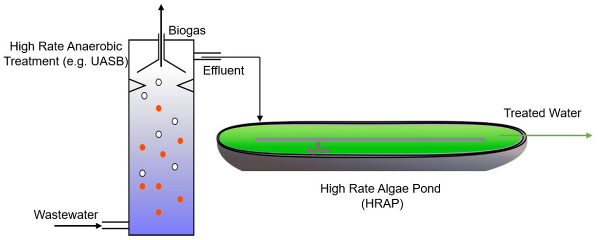

FIGURE 1. **Flow diagram of a two stage wastewater treatment process. In first stage, wastewater passes through the UASB reactor where organic waste is decomposed, mineralising nutrients whist producing bio-gas as a by-product. While in the second stage, wastewater is passed to algae raceway, where a paddle wheel circulates its flow. This stage removes remaining organic waste and other nutrients via algal and bacterial growth.**

a skewed distribution it is challenging to achieve generalization. This problem is particularly aggravated in the wastewater industry where only few data sets are available that include non-recurring seasonal patterns and odd anomalies. These peculiarities make it challenging to divide the data set into a training, validation and test sets as trivial split in time can directly result in partitions that have different statistical distributions. Learning over such data sets may not be a true demonstration of learned model generalization. We propose a robust process for sampling small data sets.

The second contribution of this work consists of a first-of-its-kind simple intuitive data division algorithm that takes into consideration the seasonal and other data distributional problems in validating deep models. The technique, locally preserves the distribution of train, validation, and test splits and encourages higher generalization of a trained model. Algorithmically, it iteratively divides the time series in to N contiguous data splits also termed windows. These temporally aligned windows are then randomly sampled without replacement and are included in test, train or validation splits. In this manner, the data splitting algorithm strives to divide local patterns in to different splits to encourage the training of model for higher generalization. In case of fixed split, as done in literature, the training patterns can be significantly different. While, proposed technique offers a more candid evaluation of a real world case.

Our third contribution is the release of annotated real-world secondary wastewater treatment data set in public domain, and allows for the development of this and other data driven research projects within the wastewater treatment field where there is a limited number of freely available data sets. The data set was curated with a combination of online and offline data measurements. We defer further discussion about data curation setup to § V of this article. Finally, we evaluate our technique using the real-world data set demonstrating superior performance to state of the art methods.

In summary, the main contributions of this paper are as follows:

1) We develop a novel soft sensor to reliably estimate Ammonium concentration (difficult to measure process variable) using a few inexpensive sensors.
2) We propose a data division algorithm for the scant highly imbalanced time series data. This algorithm enables the evaluation of a deep model's robustness and generalization in a fair manner.
3) We release a real-world annotated data set of a secondary wastewater treatment process also known as HRAP. This data set can be a stepping stone to boost the research in challenging practical scenarios of wastewater treatment processes.
4) We provide detailed comparison analysis of proposed architecture against five previously used methods. We empirically and qualitatively demonstrate superiority of our scheme. The proposed GRUconv enables performance gain up to 37% in root mean squared error over the closest competitor.

The remaining article is organized as follows. In Section II we review the related literature. We formulate the problem in Section III. The proposed technique is discussed in Section IV-A. We present experimental setup details in Section V and results in Section VI. The article concludes and discuss some future directions in Section VII.

II. LITERATURE REVIEW

In the recent years, owing to stringent environmental regulations, there has been an ever-increasing interest for the prediction of process state variables. The relevant work has followed a bifurcated approach that includes kinetic models and soft sensors. Kinetic models [27] are based on First Principle Models (FPM) that are engineered to simulate a mathematical model of an underlying process. Due to this inherent dependence on representation, they are case-specific, do not generalize to slight modifications [28] and require experts domain knowledge for being devised. Besides, they focus on ideal steady-states of the processes that limits its adoption for practical scenarios [29]. On the other hand, soft sensing enables ease of customization to variations [23], less dependence on the domain expertise and are known for their better generalization.

Soft sensors have been devised for a broad range of applications in the wastewater industry, using modelling techniques, such as PCA [30], PLS [31], SVMs [32], and ANNs [33]. Among these Contemporary methods ANNs with Feed Forward Neural Network (FFNN) architecture being the most popular choice [2], [34]. These models have been explored to predict Biochemical Oxygen Demand (BOD) [35], Suspended Solids (SS) [36], nutrient removal [14], and a range of other processes variables [1], [15]. Recently, [10] has compared FFNN with support vector machines (SVMs) to predict total Nitrogen concentration. Interestingly, the study has highlighted that SVM models outperform FFNN models. Despite the reasonable performance of FFNN and SVM, they cannot inherently capture the temporal notion of processes and assume all samples are independent. However, the industrial processes have dynamical nature where the process data has temporal correlations [37].

These temporal patterns have inspired the usage of recurrent neural networks (RNNs) [38].

Only a handful of studies have explored RNNs [21], [22], [39], [40]. Bhattacharjeeet and Tollner proposed RNN based approach to predict BOD and nitrate [40]. In [21], a particular variant called Long Short Term Memory (LSTM) has been demonstrated to forecast ammonium and total nitrogen concentrations. While Mamandipoor *et al.* [41] has leveraged a fully automatic stacked LSTM network for fault detection. Cheng *et al.* [23] has provided a detailed analysis of recurrent networks that includes GRUs, vanilla-LSTMs and its flavours for the prediction of process variables. These studies have demonstrated that deep learning models are capable to provide higher degree of accuracy. Based on their success, we now witness its wide adoption for other prediction tasks such as trajectory forecasting [42], process control and automation [43], etc.

For the soft-sensing problem, we propose a novel ensemble approach that combines GRUs and CNNs to approximate process primary variable using easy to measure variables. Our method first utilizes GRUs to extract global features of a time series while preserving the temporal history of samples. These temporally rich features are passed onto CNN that extracts local dependence in the data for prediction purposes. Our selection of GRUs is motivated by the known vanishing and exploding gradients [44] issues of RNNs. The detailed empirical evaluation in § V establishes the superior performance of the proposed scheme against other approaches. In our evaluation, we consider the data issues prevailing in the HRAP data set, and we carefully prepared the data for training using data division algorithm 1. We will release our data division algorithm and GRUconv code, along with annotated real-world secondary wastewater treatment data set in public domain to support the paradigm of reproducible research and for the benefit of research community.

III. PROBLEM FORMULATION

In this section, we formulate the problem of soft sensing. Let us suppose that we have a set of time-series $[p_0, \ldots, p_n, q_0, \ldots, q_m]$ which is acquired from different physical sensors on the treatment plant. Here, we have deliberately used different symbols to indicate 'easy to measure' variables as q, and 'hard to measure' as p. We would like to predict each of the series $[p_0, p_1, \ldots, p_n]$ from $[q_0, q_1, \ldots, q_m]$.

For each i, suppose

$$p_i = [p_i^{\langle 0 \rangle}, \ldots, p_i^{\langle T \rangle}], \tag{1}$$

$$q_i = [q_i^{\langle 0 \rangle}, \ldots, q_i^{\langle T \rangle}], \tag{2}$$

that are sampled at a common interval denoted by T. Given, $\bar{q} = (q_0, \ldots, q_n)$ is a subset of easy to measure variables, we are interested in estimating a prediction function h_i for each p_i such that,

$$\nu_i^{\langle t \rangle}(\psi) = h_i(\bar{q}^{\langle t-W \rangle}, \ldots, \bar{q}^{\langle t \rangle}; \psi), \tag{3}$$

where $W \in \mathbb{N}^+$ is the window of lagged (past) observations, $\nu_i^{\langle t \rangle}(\psi)$ approximates $p_i^{\langle t \rangle}$ and is parameterized by ψ. These parameters are stochastically estimated by casting our goal as an optimization problem to minimize an error function. We define the error as the mean absolute distance between the predicted value and the ground truth as,

$$\delta_{\mathrm{i}}(\psi) = \frac{1}{T-W} \sum_{j=W}^{T} |(\nu_i^{\langle j \rangle}(\psi) - p_i^{\langle j \rangle})|. \tag{4}$$

The soft sensing problem for p_i is then to solve,

$$\psi^* = \min_{\psi} \left[\delta_i(\psi) \right]. \tag{5}$$

The above problem can be solved efficiently by a variety of gradient descent algorithms, we defer the discussion of particular choice to §V.

IV. PROPOSED APPROACH

We first provide a discussion of the algorithm to prepare test, train and validation splits from a limited data and then provide detailed analysis of our proposed neural network architecture.

A. DATA DIVISION

As discussed in § I, it is imperative to understand the effect of fixed partitioning when we are restricted by the number of training samples. Generally, when a data set is small, it is prone to have an unbalanced statistical distribution. A simple division of such data into training, validation and test splits can result in significantly different distributions in the sampled training, validation and test sets. That in turn cast a difficult learning objective that potentially leads to poor model generalization. It is known that machine learning techniques generalize well when there is a similar statistical distribution of data in the training, test and validation data splits. In order to make best use of data and circumvent these statistical data distribution issue, we propose the Data Division Algorithm 1.

The abstract concept of the algorithm is intuitive. It takes window size $\mathcal{W}$, data samples $\mathcal{D}$, and train/validation ratio as input and returns the partitioned data respectively. On *line* 1 we first initialize $\mathcal{D}_{train}$, $\mathcal{D}_{val}$ and $\mathcal{D}_{test}$ to an empty set and compute number of possible steps ($\mathcal{M}$) from given data samples $\mathcal{D}$ and window size $\mathcal{W}$. These steps govern the number of iterations in our algorithm. In a given iteration, it samples continuous chunk of values w from the source data D as given on *line* 8 where the cardinality of this set is fixed to $\mathcal{W}$. The values are picked based on the start and end indices computed on *lines* $3-5$. Afterwards, the algorithm decides where does the set w belongs to by sampling from a generalized multinouli distribution on *line* 10 where each category has equal probability. Based on this random selection, it places the selected window in the respective train, valid, or testing set as given on *lines* $11-22$ and increments the loop iterator. The process is repeated until the maximum number of iterations have been reached. Thus, in contrast with fixed partitioning, this approach encourages a similar patterns

Algorithm 1 Data Division Algorithm

```
Input: Window Size $\mathcal{W}$, data samples $\mathcal{D}$, training ratio $\mathcal{N}_{train}$, validation ratio $\mathcal{N}_{val}$
Output: $\mathcal{D}_{train}, \mathcal{D}_{validation}, \mathcal{D}_{test}$.
Initialize $\mathcal{D}_{train}$ to {}, $\mathcal{D}_{val}$ to {}, $\mathcal{D}_{test}$ to {}, window steps $\mathcal{M} = \lfloor \frac{|\mathcal{D}|}{\mathcal{W}} \rfloor$, $\mathcal{M}_{train}$ to $\mathcal{M} \times \mathcal{N}_{train}$, $\mathcal{M}_{val}$ to $\mathcal{M} \times \mathcal{N}_{val}$, step $m$ to 0, start index $u$ to 0, end index $v$ to 0, choice $c$ to 0.
while $m < \mathcal{M}$ do
    $u \leftarrow m \times \mathcal{W}$
    $v \leftarrow (m+1) \times \mathcal{W}$
    if $v > |\mathcal{D}|$ then
        $v \leftarrow |\mathcal{D}|$
    end if
    $w \leftarrow \mathcal{D}[u \dots v]$, s.t $|w| = \mathcal{W}$
    $\mathcal{D} \leftarrow \mathcal{D} - w$
    $c \sim$ Categorical$(K = 3, p = 1/3)$
    if $|\mathcal{D}_{train}| < \mathcal{M}_{train}$ and $c = 1$ then
        $\mathcal{D}_{train} \leftarrow \mathcal{D}_{train} + w$
        $\mathrm{m} \leftarrow \mathrm{m} + 1$
    end if
    if $|\mathcal{D}_{val}| < \mathcal{M}_{val}$ and $c = 2$ then
        $\mathcal{D}_{val} \leftarrow \mathcal{D}_{val} + w$
        $\mathrm{m} \leftarrow \mathrm{m} + 1$
    end if
    if $|\mathcal{D}_{test}| < \mathcal{M}_{test}$ and $c = 3$ then
        $\mathcal{D}_{test} \leftarrow \mathcal{D}_{test} + w$
        $\mathrm{m} \leftarrow \mathrm{m} + 1$
    end if
end while
return
```

to be included in test, train and validation splits as shown in Figure 2.

The similarity of distributions in sampled sets of data, i.e., train, validation and test can be analyzed by computing the statistical distance between the available dataset and individual splits. Kullback-Leibler (KL) divergence [45] is a known technique in machine learning community that computes the relatively entropy to estimate differences in distributions. Figure 2 summarizes the KL divergence of fixed partitioning and Algorithm 1 data partitioning. It is evident that fixed (naive) partitioning of data results in higher KL divergence score between train, validation and test sets. In addition it is also worth noting that test and validation have a very low dynamic range in comparison with original data. This may result in overestimation of performance for some chunks but may not necessarily lead to generalization. It is also evident that Algorithm 1 data partitioning results in significantly lower KL divergence score that indicates higher similarity among these sets and therefore highlights the suitability of proposed technique in data partitioning.

For the soft sensing problem, as highlighted in the literature review in § II that most popular approaches utilize statistical techniques with the recent shift towards the data driven neural network models. Our work focuses on deep learning techniques as they are easy to scale and have demonstrated superior performance in variety of applications [46].

B. GRUconv

Existing work has primarily taken a bicephalous approach, where one stream has focused over exploiting global patterns by leveraging the sequential processing models such as Recurrent Neural Networks (RNNs), while the other stream has focused on the local patterns extracted via CNNs to make predictions. There is need to combine global context alongside localized features to achieve enhanced soft sensing ability. In this work, we seek to mitigate this deficiency and propose GRUconv. It is a novel hybrid algorithm that utilizes GRUs in combination with CNNs to encode both, short term and long term representations of the time series. Figure 3 presents schematic of our hybrid GRUconv model.

Gated Recurrent Unit is a modification to RNN that allows to deal with long-range sequences. It alleviates the vanishing gradients issue by introduction of hidden state update and reset. This allows it to filter and retains relevant information. GRU uses two gates, namely reset (Γ_r) gate and update (Γ_u) gate. The (Γ_u) gate determines how much of the past information is needed for the next state, and the (Γ_r) gate decides how much of the previous memory to forget.

Our hybrid model first computes the hidden state $h^{\langle t \rangle}$ of a recurrent unit i.e., GRU at time step t on multivariate input X as:

$$\begin{aligned} \Gamma_u &= \sigma(W_u[h^{\langle t-1 \rangle}, X^{\langle t \rangle}] + b_u), \\ \Gamma_r &= \sigma(W_r[h^{\langle t-1 \rangle}, X^{\langle t \rangle}] + b_r), \\ \bar{h}^{\langle t \rangle} &= \mathrm{ReLU}(W_h[\Gamma_r \odot h^{\langle t-1 \rangle}, X^{\langle t \rangle}] + b_h), \\ h^{\langle t \rangle} &= \Gamma_u \odot \bar{h}^{\langle t \rangle} + (1 - \Gamma_u) \odot h^{\langle t-1 \rangle}, \end{aligned} \tag{6}$$

where, $\odot$ is a element wise product, $\sigma(.)$ denotes *ReLU* activation, $W_{u/r/h}$ are learn-able network weights, and b_j, $\forall j$ are the respective biases.

The output of GRU i.e. hidden state $h^{\langle t \rangle}$ captures the global features from the input sequence at each time step and is reshaped into a tensor of target shape 20 × 20, and then simultaneously fed into a 1D convolutional layer. Each convolution layer has multiple filters of size w also known as kernel/window. These kernels sweep through the $h^{\langle t \rangle}$ and performs a convolution operation as:

$$h_\varsigma^{\langle t \rangle} = \sum_{j=1}^{w} \sigma(W_j * h^{\langle t \rangle} + b), \tag{7}$$

where, $*$ denotes the convolution operation between the j-th, W_j filter of size w and input $h^{\langle t \rangle}$. Here, b is the kernel bias and σ is a *ReLU* activation function.

Convolution is an iterative process where in each step convolutional kernel (W_j) is sweeped over the input $h^{\langle t \rangle}$ such that output is the sum of kernel filter multiplied with the corresponding input values. Each convolution filter produces

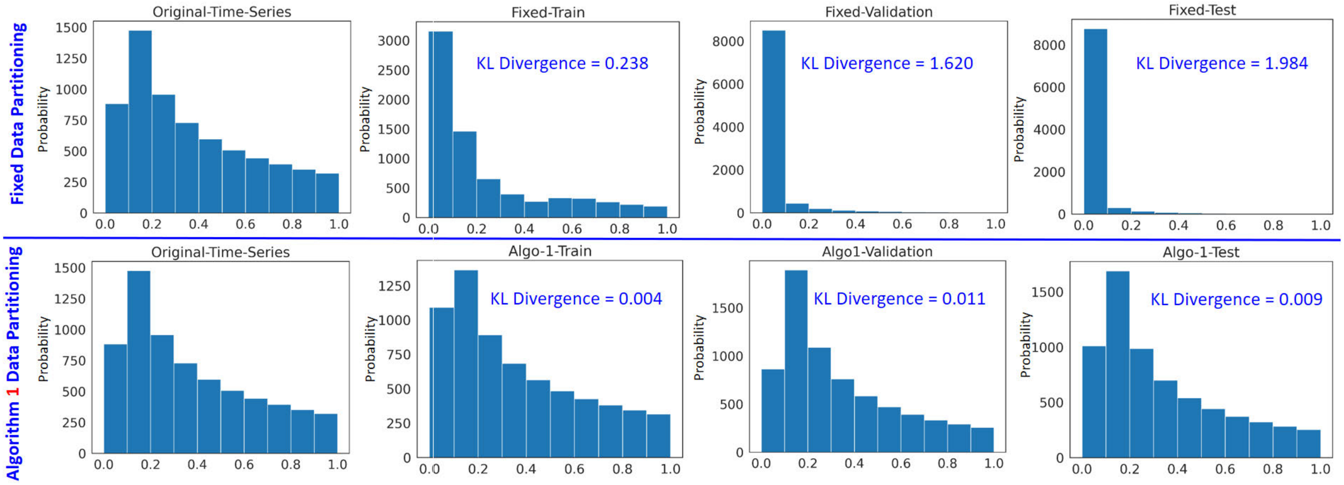

FIGURE 2. Kullback–Leibler (KL) divergence to measure the difference in the statistical data distribution when partitioned using Fixed Partitioning and Algorithm 1. The lower KL divergence indicate similarity between their distributions, and the lower KL divergence can be seen in Algorithm 1 data partitioning.

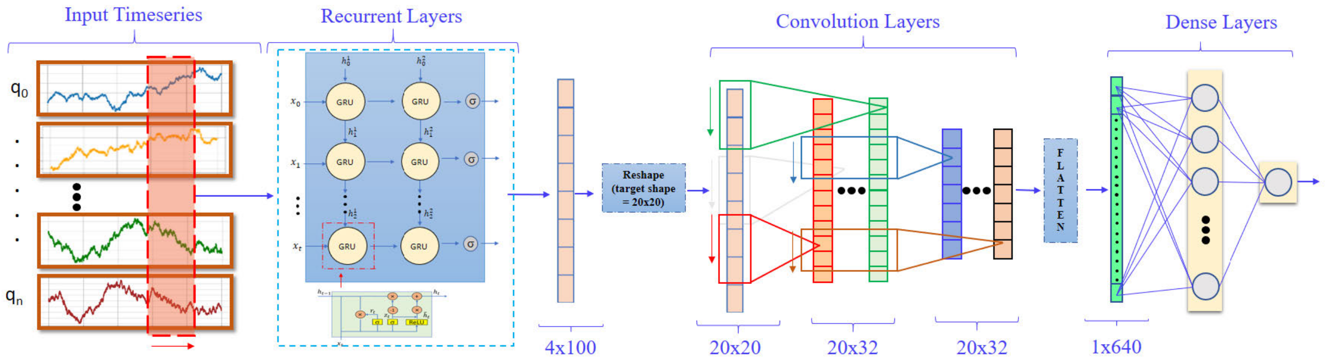

FIGURE 3. Schematic overview of our proposed technique. A multivariate time series of an easy to measure variables highlighted as $(q_0, q_1, \ldots, q_n)$ are first processed by GRU layers to encapsulate temporal dynamics enriched information. Afterwards, these temporal features are reshaped into a target shape of 20 × 20 tensor, and then fed to cascaded 1D convolutional filters to incorporate localized information. The resultant maps are flattened and fed to a densely connected layers for the final prediction. Different layers and their respective output feature size has been indicated in blue fonts.

a separate output sequence, thus number of filters and kernel size decide the output feature size. These layers offer several discriminative properties: sparse interaction, and parameter sharing [47]. As the convolutional filters sweeps over the input time series it captures local information on various scales. This implies that it requires very few parameter to learn kernel weights and this is known as sparse interaction. Also, when the same filter sweeps across all the time steps it shares its weight and this property is called parameter sharing. The sparse connectivity and parameter sharing significantly reduce the number of learnable parameters due to their shared nature. This in turn, improves the efficiency of learning in comparison to recurrent layers. These discriminative resultant features are then flatten and finally processed by dense layers to predict the hard to measure variable.

The proposed hybrid GRUconv model benefits from the light weight convolution layers in terms of computational efficiency as well as its refinement of temporal features through utilization of local correlations. The results indicate that the proposed algorithm provides highly competitive results in both quantitative and qualitative aspects. We have provided a detailed discussion over the evaluation in Section VI and demonstrated that proposed architecture is able to learn both long and short term trends that is reflected by its superior performance.

V. EXPERIMENTAL SETUP

We first give details of the data set, its curation, and data pre-processing pipeline before discussing the experimental setup details.

A. HRAP-DATASET

Algae raceways are also known as High Rate Algae Ponds (HRAPs). HRAPs are a secondary wastewater treatment process that removes nutrients and organic waste via algal and bacterial growth. The HRAPs are fed form a high rate anaerobic digestion process known as an Upflow Anaerobic Sludge Blanket digestor (UASB), which acts to degrade particulate organic matter to release nutrients and organic matter for further treatment in the HRAPs. Physical sensors are deployed in algae ponds for monitoring process variables including pH, temperature, dissolved oxygen and turbidity and nutrient (ammonium and nitrate) concentrations. Among these process variables, the concentration of Ammonium is of significant importance, it is a prime indicator of treatment efficiency and is highly regulated in discharge water.

The data set was acquired from one continuous algae raceways (also known as HRAP) and weather station located at the Luggage Point sewage treatment plant in Pinkenba, Queensland (as courtesy of Urban Utilities). The online HRAP measurements (15 minute logging interval) were collected via Xylem WTW VARiON®Plus 700 IQ Ammonium and Nitrate, pH, Dissolved Oxygen, Turbidity, Total Suspended Solids and Oxidation Reduction Potential Sensors. Meteorological observations were collected at 15 minute intervals using a Vaisala WXT536 weather station (temperature, humidity, pressure, wind and rain) and a Global Water WE300 solar radiation sensor. Offline analytical laboratory measurements were taken 2-3 times a week, including Ammonium (NH_4^+) and nitrate concentrations which were measured using a Lachat Quick-Chem 8000 Flow Injection Analyser (Lachat Instrument, Milwaukee, Wisconsin). To ensure consistency of the collected online data, pH, turbidity and dissolved oxygen probes were calibrated at monthly intervals using appropriate standard solutions and the ammonium and nitrate probe was calibrated using grab samples to assess both the nutrient levels and water matrix, using analytical methods described. This allowed for verification of the logged readings and where substantial probe drift was observed this data was removed form the analysis.

The deployed physical sensors sample the process variables every 15 minutes and log them into the database. We have utilized the data that was logged between 25 July 2020 and 7 July 2021. Table 1 describes the sensors and their details that have been used to curate time-series data utilized in this work. Along with the above mentioned time-series data (see Table 1), we have explicitly modelled time of the day (ToD) to segregate the night and day time readings of the sensor. For this purpose, we utilized a ramp function whose value gradually increases every 15 minutes and once it reaches the day end, it drops to zero. The modelled time and easy to measure process variables are then used as an input for training of our data driven models. The objective of our model is to approximate the Ammonium sensor with sufficient accuracy to make the \$10K sensor redundant.

Figure 4 (see § VI) highlights the diverse scales in the original time-series of ammonium. We bound the diverse range of the time series by performing 'min-max' normalization that in turn is vital for the training of deep models. Further, the acquired data from the wastewater treatment plant has a duration that is roughly a year, thus it essentially lacks recurring seasonal patterns. Machine learning models generalize well when there is a similar statistical distribution of data in the training, test and validation data splits. All these considerations make the development and learning of a data-driven model on such a data set a challenging task.

B. DATA PRE-PROCESSING AND EVALUATION METRICS

Synthetic data is usually free of any noise and therefore it is often utilized for training purposes. However, real life data suffers from a range of external perturbations that results in noisy [48] or missing entries [49]. Presence of noise in real data causes the deep learning models to learn an incorrect representation and this results in sub-optimal outcomes. Thus pre-processing of data is an essential step in training data driven models.

C. MISSING ENTRIES

K-nearest neighbours algorithm is widely used to impute the missing entries [49]. However, they may add process dependent biases in the training data. We have therefore adopted a conservative approach and resorted to remove all the entries where any sensor data was missing in the available data. After removal of missing entries, we have $25,812$ samples in total.

D. NOISE REMOVAL

For noise removal several approaches have been proposed. For example, in [23] exponential smoothing filter was used to de-noise real world WWTP data. Similarly, we followed the popular approach *smoothdata* and utilized its implementation available in Matlab [50]. In essence, this technique has a moving average filter that estimates the window size heuristically, such that it maintains x percentage of input energy.

All the input/output features were normalized to [0,1]. The normalization enabled us to fairly compare prediction results in a scale independent manner.

E. EVALUATION METRICS

We evaluate the performance using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2). For details on advantages and disadvantages of each metrics, we refer readers to [51] paper.

F. DATA SET PARTITIONING

The empirical evaluation includes the HRAP data set that has been discussed in Section V-A. The available data of $\approx$ 12 months includes seasonal variations and other non-stationary trends. It is not trivial to divide such data set into a training, validation and test sets since a naive split in time can directly result in partitions that have different statistical distributions. This may result in under or overestimation of the trained model. To circumvent this issue, we prepare data divisions using Algorithm 1. It create windows of 24 contiguous data samples. These windows are then randomly sampled (80%) to create the training split, while the test and validation split includes 10% each. Following the machine learning norm, the model that had the best score over the validation split was saved. Further, its score over the test split was then evaluated and reported.

TABLE 1. List of relevant physical sensors for wastewater monitoring. The first column indicate the sensor name, the second column provides the measurement units and the last column indicates the instrumentation cost. It is evident that Ammonium and Nitrates are associated with very high sensing costs.

Physical Sensors	Description (units)	Instrumentation Cost ($\approx$AUD)
Galvanic Dissolved Oxygen	Measurement of Dissolved Oxygen (mg/L)	\$ 2000
pH	Single-rod measuring cell for wastewater (-)	\$ 1500
Turbidity	Scattered Light Measurement (NTU)	\$ 3500
Temperature	Algae pond's temperature (°C)	Included with other sensors
Ammonium and Nitrate	Ammonium and Nitrate concentrations (mg/L)	$\geq$ \$ 10,000
Rain	Rain intensity (mm)	\$ 2000

G. MODELS TRAINING

In our experiments, we employ CNN, LSTM, GRU, Bi-GRU, Bi-LSTM and GRUconv for the regression task. We observed that our framework performs similar for the choice of history samples between 4 and 8. In this study, we have fixed the size of history as 4 samples (1 hour) for every input feature. All models are trained with Keras package [52] on top of TensorFlow [53] in python. Maximum epochs were set to 200 with early stopping mechanism. Adam [54] optimizer was selected for the learning of model parameters with a dropout probability of 0.2. We set learning rate between 10^{-4} to 10^{-3} to train the models. We set the batch size of 32 and use NVIDIA GeForce RTX 3080 GPU to run our experiments. Our code implementing data division algorithm, GRUconv architecture, pre-trained models and HRAP data set will be released on acceptance of this research paper from https://github.com/MairaAlvi/AmmoniumSensorApproximator.

H. MODEL PARAMETERS

The number of parameters of a neural network model governs the performance, its memory foot print as well as the run time efficiency of an algorithm. To ensure fair comparison to promote reproducible research paradigm, we have included model parameters in the Tables $1-6$ of the Supplementary Material. All of the models were trained in a similar fashion that includes fixed batch size of 32, dropout probability of 0.2, and learning rate of .0010. Please note that for all the models, the best hyper-parameters were identified by varying the number of layers between $2-6$, the learning rate between 10^{-4} to 10^{-3}, dropout probability between $0.1-0.5$. Similarly, several choices of activation function were explored. This includes 'ReLU', 'Tanh', 'SELU' as well as no activation. In addition, the number of neurons in different layers, including recurrent layer and convolutional layers were also varied in a broad fashion. The reported results include the best performing representative model.

VI. RESULTS AND DISCUSSION

We have conducted a detailed empirical study over the HRAP data set to investigate the performance of our proposed technique. For comparison purposes, the accuracy of each algorithm is measured over popular metrics including RMSE, MAE, and R^2. The best results are summarized in Table 2. To avoid the impact of random initialization [55], the reported scores are averaged over 10 runs. The Table 2 highlights that proposed technique of GRUconv outperforms all other methods and achieves the lowest 0.03062 value of RMSE, which provides a $\frac{0.04862-0.03062}{0.04862} \times 100 = 37\%$ gain over the closest competitor. GRUconv maintains superior performance over the other metrics with a gain of 23% for MAE, and 4% for R^2. We conjecture that our approach's main strength is attributed to its superior feature extraction that extracts both, macro and micro temporal information, in contrast with other methods. GRUconv encodes global dynamics of a time series via temporal modelling through the GRU layers and afterward enhances it by repeated application of convolution filters. Finally, these features are weighted to produce final regression value.

TABLE 2. Comparison of techniques over different evaluation metrics. The first column indicates the description of training algorithm. The rest of columns indicates the average error alongside standard deviation over training 10 models from scratch on a single sampled data. The error metrics include RMSE in second column, MAE in third column, and R^2 in last column. The results in bold highlight the best performing model scores.

Model	RMSE	MAE	R^2
Stacked CNN	.05717 ± .0258	.0236 ± .0082	.9189 ± .0607
Stacked LSTM	.05243 ± .0188	.0182 ± .0050	.9321 ± .0474
Stacked GRU	.04862 ± .0143	.0164 ± .0033	.9429 ± .0360
Stacked Bi-GRU	.05212 ± .0159	.0179 ± .0061	.9399 ± .0272
Stacked Bi-LSTM	.06252 ± .0206	.0199 ± .0060	.9090 ± .0446
GRUconv	**.03062 ± .0072**	**.01261 ± .0031**	**.9791 ± .0095**

Although quantitative evaluation is important to measure the performance of a scheme, a qualitative study is vital to analyze the performance of a scheme at specific regions such as flat, spikes or crevices. We have included qualitative comparison of different techniques in Figure 4 that reveals several interesting insights. It can be observed that, CNN predictions suffer from significant temporal misalignment that exhibits severe oscillatory behaviour between days 15 to 30, 80 to 90, and 150 to 170. However, its predictions on the other regions of the time series are relatively reasonable. We conjecture that the transition of ammonium time series for the above mentioned days may be improved via inclusion of a global context available in the previous time steps. CNN utilize local features over a narrow input window for decision purposes. Although they can capture local features, the inherent absence

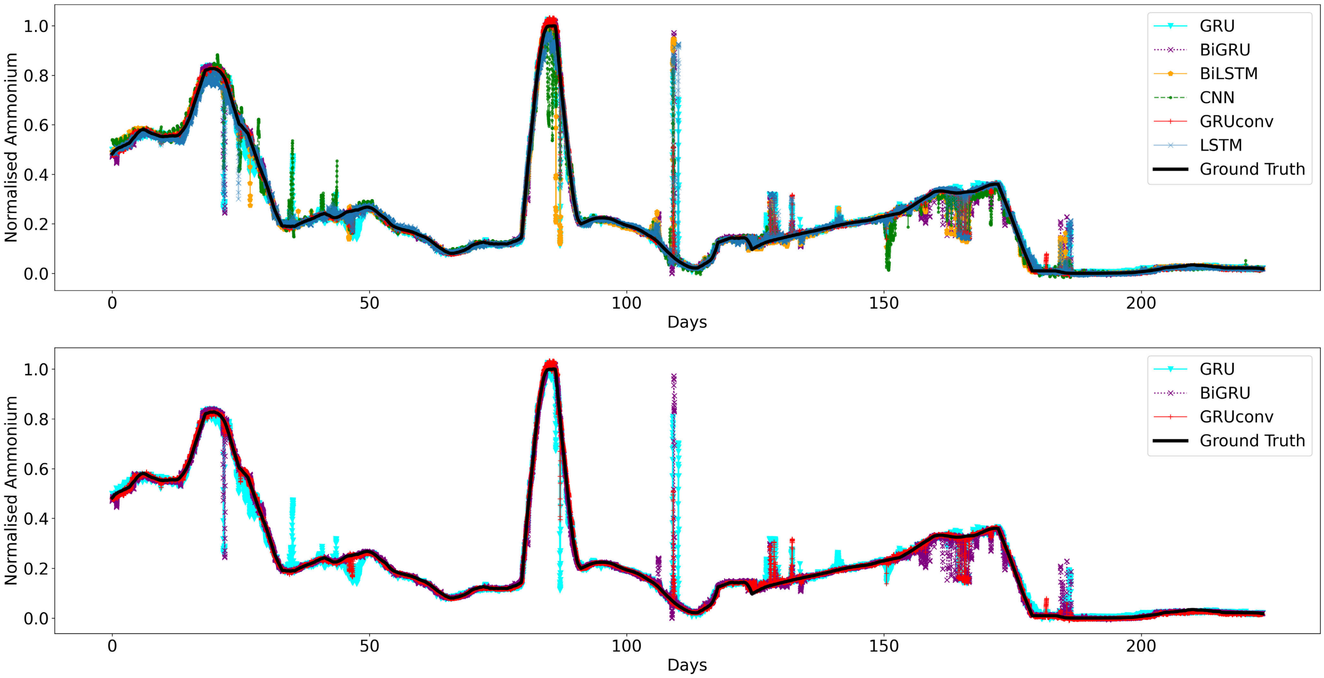

FIGURE 4. **Qualitative comparison of predictions by different deep learning algorithms. Proposed algorithm (GRUconv) results are indicated as red line, while the reference Ground truth values are indicated in black. The top subplot compares the performance of each representative models, while bottom subplot compares GRUconv with its closet competitors (BiGRU and GRU) for clear visualization. Note: this figure is best visualized in the digital format.**

of memory limits their ability to capture a global context to perform well in challenging wastewater prediction.

We turn next to recurrent neural networks that are specialized for processing sequential data [47], [56]. It can be observed that its variants including GRU, LSTM, Bi-GRU, and Bi-LSTM often suffer from temporal misalignment and overshooting. LSTM predictions follows correct shape of ground truth time series, but with some temporal misalignment on days 11, 18, 106 to 108, and on day 180. It underpredicts with an error of approximately 30% on day 11, and regions between days 106 to 108 are over predicted with a margin of 70%. While GRU predictions are precise and temporally localized except for day 18, 30, 80, 102, 180 and from 110 to 115. Both, LSTM and GRU provides adequate approximation of the temporal variations in ammonium. However, certain regions in the ammonium time series are either under or over predicted that may provide false information to the plant operator. Reliable and robust estimates of processes is vital for the process management and control of wastewater treatment plants.

The recent variants of RNNs includes the bidirectional processing units and have substantially outperformed unidirectional neural networks in several applications like speech recognition [57], and hand-writing recognition [58]. In terms of operations the information flows forward and backward through time. So effectively, one RNN moves forward from start to the end of sequence, and another RNN moves backward through time (i.e., from the end of sequence). In context of wastewater treatment plant, the community has adopted it for p-step ahead prediction of flow rate [42] with significant improvement over other methods. However, in context of discussed application, our empirical and qualitative results unfolds a different story. We have included additional comparisons with the bidirectional models to validate the superior performance of GRUconv. Despite highly competitive R^2 score of Bi-GRU, the model's predictions are imprecise with a margin of 60% at day 18, 20% at day 80, 80% at day 102, and 18% between days 160 to 180.

A close inspection of qualitative results, indicates that Bi-LSTM predictions are non-sharp, with possibly large temporal misalignment. It is emphasized that, wastewater treatment plant is a causal system, and any information of the system at time t is only dependent on the information from the past therefore the features from future to aid the previous time steps as done in bi-directional recurrent neural networks may not be helpful and may potentially harm the generalization ability. In contrast, our method embeds both, global and local temporal dynamics by hierarchically applying GRU and CNN respectively. As illustrated in Figure 4, it has some temporal misalignment's at day 40 and 175, but the error margin is less than 5%. It also over-predicts by an error of 30% on day 102, and 8% on days between 130 to 135. Overall, it demonstrates precise temporal localization throughout the different regions of the time series. In Figure 5, we include the visualization of R^2 (correlation of coefficient) score for top four best performing algorithms over different train-test splits (for cross-validation) of available data. These splits were controlled by setting random sampling in Line 10 of

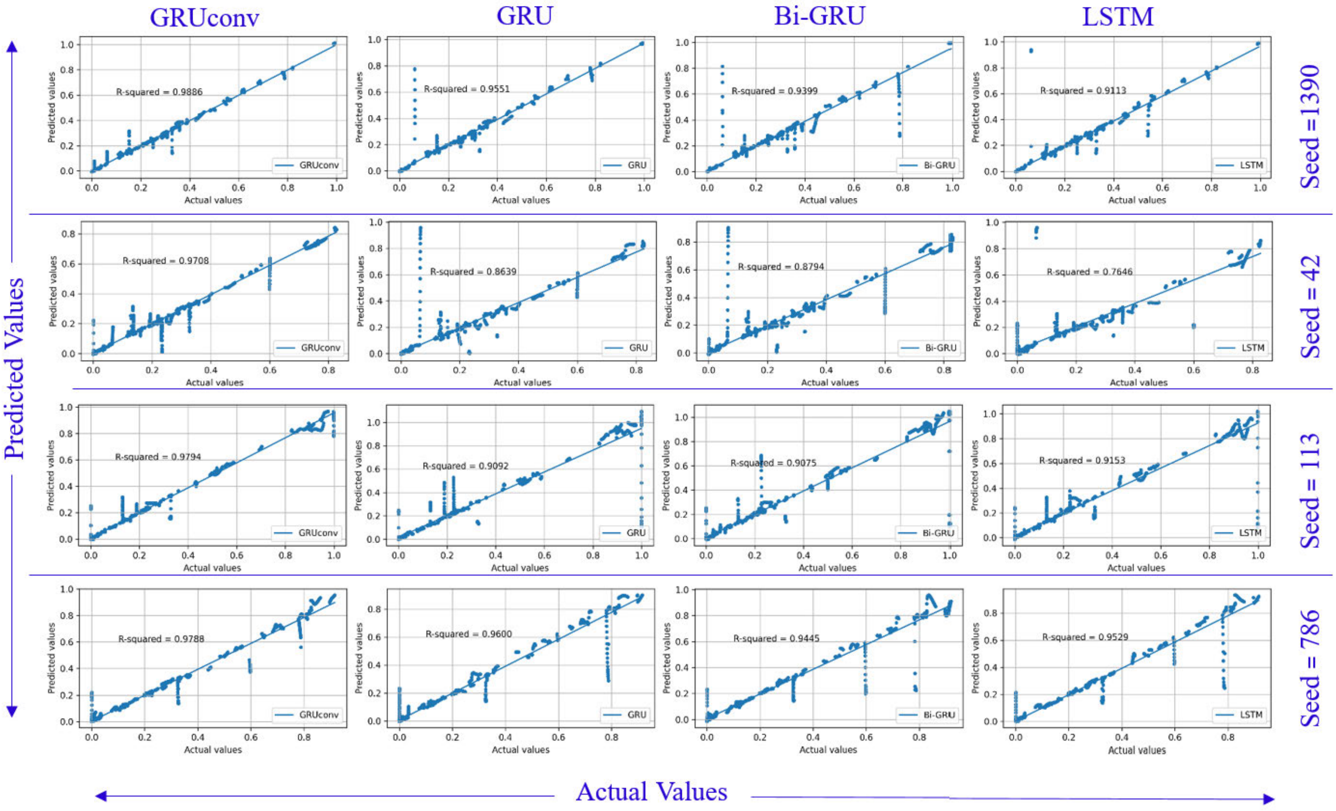

FIGURE 5. Coefficient of determination (R^2) for different algorithms over different train-test splits of data sampled by random seed. The algorithm is indicated in the top row and the seed values are indicated in the last column respectively. These correlation of coefficients indicates match between model predictions and the ground truth, where higher scores imply a better fit. For brevity, only top four best performing model scores have been included.

Algorithm 1. In practice it is achieved by setting the seed value of the sampling algorithm. R^2 scores illustrate how well the respective model predictions approximate to the true values. As can be seen, our method outperforms other methods by a significant margin. This justify the noticeable gain achieved by the proposed architecture over the state-of-the-art methods.

A. FAILURE CASES

One important aspect of analyzing the algorithm's performance is to consider the failure cases of poor predictions. Figure 4 shows that some days are consistently hard to predict for all the models including GRUconv. The predictions of the top 3 performing models and ground truth for reference are shown in Figure 6 and Figure 7. As can be seen, in Figure 6 on days between 108.75 to 109.25 Bi-GRU, GRU, and GRUconv suffer from overshooting with a magnitude of 90%, 80%, 50% error respectively. This is anomalous behaviour. For the remaining days GRUconv predictions are close to the groundtruth. We inspected the input features and noticed that during the days with anomalous predictions a wet weather event was reported with a maximum rainfall accumulation between 27 − 40 *mm*. The rainwater associated with the wet weather events dilutes nutrients which in turn affects pH and turbidity. Consequently, the pH reduces more than usual and turbidity also drops due to dilution and limited sunlight associated with the wet weather event.

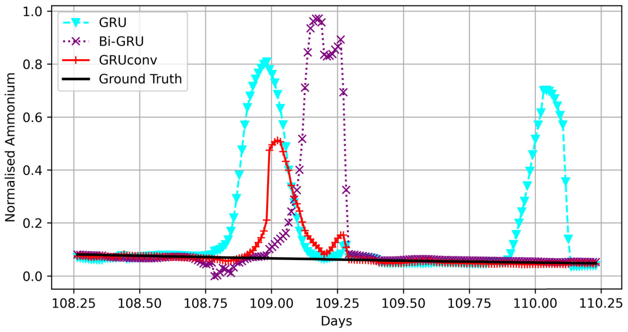

FIGURE 6. Illustration of qualitative results of hard to predict days by all the models (For brevity only top 3 best performing models are plotted). Each model overshoot between day 108 to 109, and GRU model predictions are also anomalous between the end of day 109 to quarter past day 110. Overall, GRUconv models predictions in comparison to GRU and Bi-GRU are less erroneous.

The soft sensor models approximate an algal system where the processes are dominated by pH and turbidity. These models are unable to predict accurately on rainy wet weather days. All models predict more (i.e. overshooting) ammonium, while in reality, it does not change much due to the dilution of the nutrients by the rainfall event. The poor behaviour

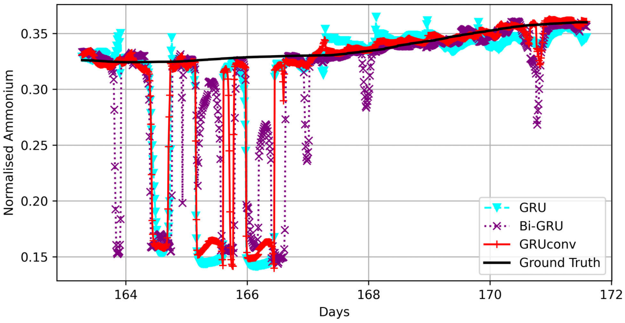

FIGURE 7. **Qualitative analysis of anomalous days that are hard to predict. All the models under predict between days 164 to 166 with roughly same magnitude of error. Alongside, Bi-GRU predictions are also imprecise before day 164, on day 168, and 170. Overall, both GRU and GRUconv have highly competitive performance on these days.**

of models on these days can be attributed to the low number of training samples for the extreme wet weather days. This in turn highlights the importance and need for larger and more comprehensive wastewater data sets. Alternatively, wet weather events could be addressed using oversampling techniques. Interestingly, among all the models GRUconv prediction had a relatively smaller overshoot and thus indicates superior performance in the challenging scenario of infrequently observed wet weather events.

Furthermore, Figure 7, illustrates additional outlier days that are not predicted well by any of the models. As shown, all the models are under-predicting with approximately 17% of error. To investigate the cause of this under-prediction, we look at the model inputs and observe that pH is overshooting these days and hence we predict a lower nutrient concentration. It is sufficient to note that Bi-GRU model predictions are off-beam on various regions between days 164 to 172. This establishes that in the wastewater treatment process, variables are driven by the causality of the system. Therefore, bi-directional traversal between future and past confuses the model that leads to erratic model predictions.

To consolidate the effectiveness of our technique as summarized in Table 2, we additionally performed cross-validation on 10 different train, validation and test splits of data sampled by random seed values using Algorithm 1. The results of GRUconv and the best performing other approaches are indicated in Table 3 and the respective box whisker plots are indicated in Figure 8. It can be seen that GRUconv performs on-par or better than the existing state-of-the-art methods and hence is a reliable model that learns the complex global and local temporal dynamics of a ammonium time series.

In Figure 8, it can be seen, both bidirectional recurrent neural networks do not achieve low RMSE in comparison to their unidirectional variants. It is noteworthy that LSTM model is erratic with large variance. However, CNN, GRU, and Bi-LSTM have very small variance. GRUconv model has marginally higher variance than GRU but overall superior performance on 10 random sampled data distributions.

TABLE 3. **Average cross validation scores and standard deviation of different models over ten random seeds. For each seed, a different train, test and validation split is created. The scores are evaluated over different metrics that include RMSE, MAE, and R^2 metrics. Best scores are indicated in the bold font.**

Model	RMSE	MAE	R^2
Stacked LSTM	.07081 ± .0127	.02095 ± .0027	.7763 ± .0266
Stacked GRU	.06305 ± .0090	.01931 ± .0016	.8970 ± .0318
Stacked Bi-GRU	.06654 ± .0103	.02165 ± .0033	.8778 ± .0413
Stacked Bi-LSTM	.07477 ± .0073	.02259 ± .0017	.8579 ± .0286
GRUconv	**.04909 ± .0106**	**.01655 ± .0022**	**.9305 ± .0318**

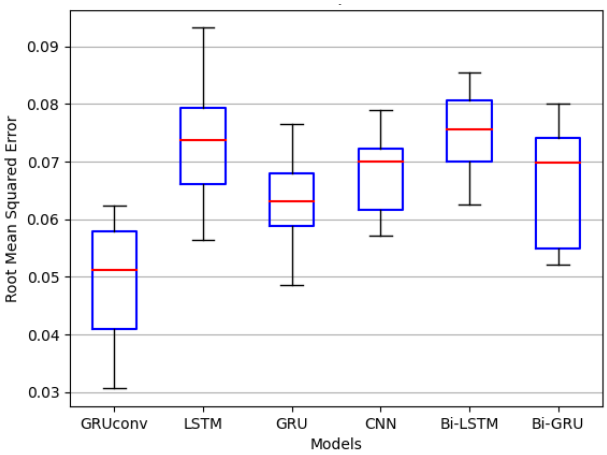

FIGURE 8. **Comparison of prediction errors (i.e. RMSE) of different models when cross-validated over 10 random sampled data distributions. Figure indicates that GRUconv's median is nearly equal to the minimum error score of GRU (closest competitor) model, thus highlighting the efficacy of our technique for a contemporary practical problem.**

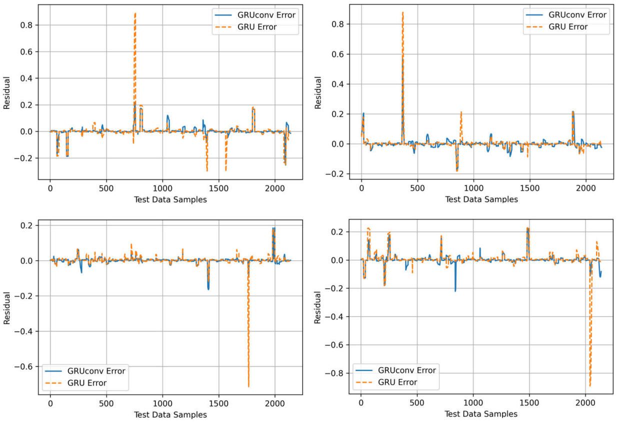

FIGURE 9. **Comparison of residuals of GRUconv and GRU (closest competitor) over four random sampled data distribution sets. In second subplot the performance of both GRU and GRUconv models is highly competitive. However, in first, third and final subplot GRUconv has minimal residual that explicitly ascertain its superior performance.**

Figure 9 shows the residuals of GRUconv and its closest competitor GRU over 4 different cross-validation sets chosen from our experiments. It can be seen that the residuals in GRUconv are consistently lower than those of GRU and that GRU has a few outlier residuals that do not occur in the GRUconv model.

When considering the use of a soft senor in an industrial process sharp changes in values are of particular concern as they are likely to lead to undesirable operational responses. In the case of process monitoring sudden changes are likely to generate false alerts, particularly out of range alerts, resulting in wasted operator time and effort. This is of even more concern if the soft sensor is to be used for process control as the sudden jump in value would generate a large instantaneous error value which may result in a large operating point change. This in turn can lead to increased operation costs, and in extreme cases can cause damage to equipment. In order for a soft senor to provide value in an industrial operational context, it needs to provide sufficient information for actions to be taken which requires both long term and short term accuracy. Therefore, our model meets the practical challenges of soft-sensing in WWTPs.

B. FEATURE IMPORTANCE RANKING

In addition to our comprehensive qualitative and quantitative analysis, we investigate the relative importance of each input feature for the predictive modelling problem. This exploration is for better understanding of the data, model, selection of optimal input features, and in the discovery of key factors in this specific domain. In this analysis, we utilized the same best performing models as indicated in Table 2. The input for each of these models was suppressed, one feature at a time. That is, for all input times each input feature was replaced by its mean value. This, allows us to keep the input shape same for the neural network models.

In Figure 10, we summarize the importance of each feature by analyzing the RMSE of the GRUconv model, where high RMSE represents the relative importance of the feature. For clarity, we have only included GRUconv results in this figure. Alongside, we observe a similar trend of feature importance for all other models (see Figure 1 in Supplementary Material). As shown in Figure 10, when pH input is suppressed, the model performance drops significantly and RMSE is ≈ 25%. This highlights that pH has the biggest impact on the prediction of ammonium. This is associated with the biological phenomenon where algal activity (and ammonia gas stripping) are related to the pH. Elevated pH coincides with high algal activity (i.e. fast carbon dioxide (CO_2) uptake). Similarly, turbidity has a significant impact on the performance of ammonium prediction. Turbidity is linked to algal biomass quantities so it represents the magnitude to which the system can respond. More algae allows chemistry in the system to change faster, with more ammonium uptake, and pH and dissolved oxygen (DO) increase.

On the other hand, DO has less impact than pH or Turbitidy. This is because DO is not as closely linked as pH to the ammonia volatilisation phenomenon or CO_2 uptake by alage biomass growth. There are other processes that will directly influence the DO concentration such as algae photosynthesis and respiration bacterial respiration, and exchange with the atmosphere. We observe that temperature and time of day have low impact on the model predictions in comparison to both turbidity and pH. Temperature does influence the treatment system activity but there were no large swings in the original temperature time series and so it does not have a big impact on the ammonia in the time frames of this soft sensor. Also, time of day, could be redundant because the pH profile reflects the time of day with its observed diurnal trend. Interestingly, rain from wet weather events has the smallest influence on the model but rain should dilute the system and change the ammonium concentration. We believe that rain has a small impact because wet weather events are infrequently observed in the data set and so the model ignores it most of the time.

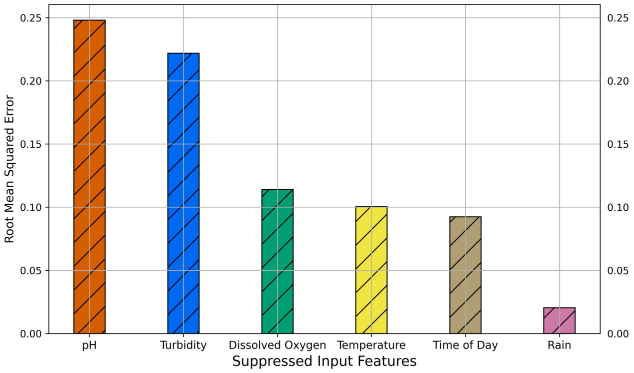

FIGURE 10. Illustration of relative importance of each feature in terms of Root Mean Squared Error (RMSE). The higher RMSE indicate more relevance of the input feature with target variable (i.e. Ammonium).

C. COMPUTATIONAL TIME AND MEMORY FOOTPRINTS

Finally, we compare the inference models for their computational efficiency, and their memory footprints of training and forward pass. The training duration averaged over 50 runs is summarized in Figure 11, and memory usage during training and forward pass are elaborated in Figure 12. In terms of computations, the CNN model is the most efficient and requires approximately 2 minutes for training. This is because the convolutional layers are easily parallelized and hence naturally suitable to hardware acceleration available in the GPUs. Next, we observe that in comparison, GRUconv takes almost twice the time as CNN's. It is inherently slow due to its sequential nature of processing time series that cannot be parallelized. As GRUconv is a combination of both GRUs and CNNs, so it is somewhat comparable to both of them. However, interestingly it is twice faster than GRUs and LSTMs due to fast convergence. Among all models, the bi-directional techniques are most expensive to train due to their sequential nature that requires dual traversals in each training example.

Memory footprint of an algorithm is important aspect to consider in real time deployment in resource constrained environments such as remote wastewater plants. It depends on several factors such as specific architectures, their ease of parallelization and implementation peculiarities. However, it indirectly indicates algorithmic complexity. We estimated the memory utilization using the Nvidia utility of '$nvidia - smi$' by running each model 10 times and taking the average.

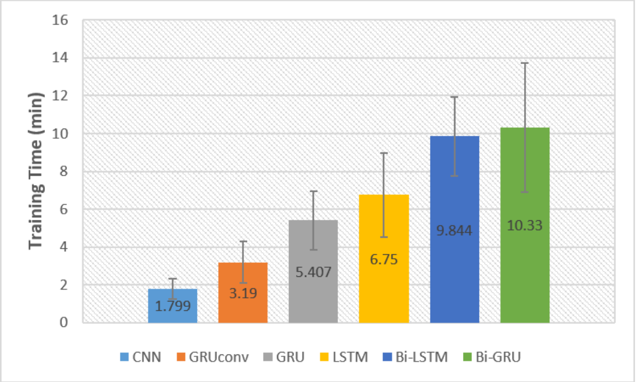

FIGURE 11. Average training duration of each model over Nvidia GeForce RTX 3080 GPU. The statistics are calculated for training 50 models over the batch size of 32, with maximum epochs of 200 with dropout probability of 20%.

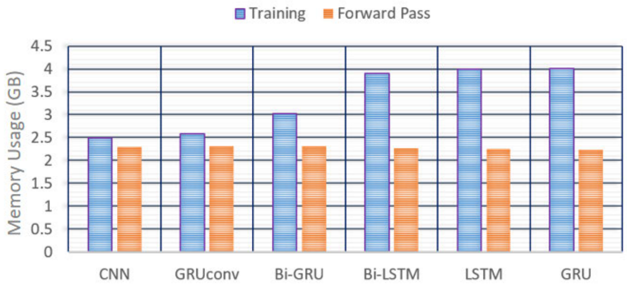

FIGURE 12. Nvidia GeForce RTX 3080 GPU memory consumption of training and forward pass, Tensorflow CNN, GRUconv,Bi-GRU, Bi-LSTM, LSTM, and GRU models with the batch size of 32, with maximum epochs of 200 with dropout probability of 20%.

The estimates values are summarized in Figure 12. It can be observed that during training, all models consumes more memory. This is because both forward and backward passes are required. However, for the predication operation in practical deployment, we only require the forward pass on a trained model which requires significantly less memory. Figure 12 indicates that there is a significant drop in memory consumption for forward pass for all uni-directional, and bi-directional recurrent layers. CNN and GRUconv have marginally low memory usage. This could be associated with the implementation ease that is associated with CNN layers. Optimized convolution operation often utilizes frequency domain via Fast Fourier Transform that is multi-fold faster than vanilla implementation. Overall, CNN and GRUconv consumes less GPU memory than other models for training and it offers similar memory consumption for forward pass.

Although the computational, and memory footprints are an important factor that governs the cost of a developed soft sensor, however it should be considered alongside the inference performance of a model. Among all techniques, GRUconv provides an acceptable balance between the quality of inference and the associated training cost. Hence, it is better suited for the soft-sensing task and can be further tested in other similar wastewater treatment parameters and processes.

The takeaway message of this Section is that the GRUconv model is obtaining the highest performance while minimizing the needed computational time and memory resources.

VII. CONCLUSION AND FUTURE WORK

We presented a novel approach to craft a soft sensor that can be a cost effective alternate to an expensive instrument in a wastewater treatment facility. Our technique performs on-par or better than the existing state-of-the-art models. Our model exploits the strength of recurrent layers to capture long-term temporal patterns and uses convolutional kernels to extract localized short-term trends for prediction purposes. The empirical and qualitative evaluation on real-world (HRAP) data set ascertain the superiority and effectiveness of our proposed GRUconv model.

Further, we proposed a data division algorithm when the time series data is not only scarce, but also lacks recurring seasonal patterns and suffers from odd anomalies. Fixed partitioning of such time series provides an overestimation of the model's performance. Our algorithm splits such data in windows to keep temporal information intact and randomly samples the windows. This results in data splits that have almost similar statistical distribution in training and testing enabling fair evaluation of the generalizability of the models. Finally, we release a real world annotated data set of a secondary wastewater treatment plant located at Luggage Point, Queensland, Australia. The public release of such data set will support the paradigm of reproducible research, for the benefit of wastewater research community.

This work inspires several promising directions for future work. Firstly, we intend to extend this idea for other wastewater treatment process variables such as Chemical Oxygen Demand (COD) [59] and micro-algal cell counts [60] that require offline laboratory analysis. Real-time estimates of these variables can enhance wastewater treatment efficiency and management. Secondly, we aim to use Bayesian deep learning [61] to investigate for any further refinements in predictions. Thirdly, new loss functions can be explored to improve learning performance. For example, explicitly utilizing silhouette information or using an ensemble of losses can be further investigated to improve qualitative and quantitative aspects.

REFERENCES

[1] G. Wang, Q.-S. Jia, M. Zhou, J. Bi, J. Qiao, and A. Abusorrah, ''Artificial neural networks for water quality soft-sensing in wastewater treatment: A review,'' *Artif. Intell. Rev.*, vol. 55, pp. 1–23, Jun. 2021.

[2] H. Haimi, M. Mulas, F. Corona, and R. Vahala, ''Data-derived soft-sensors for biological wastewater treatment plants: An overview,'' *Environ. Model. Softw.*, vol. 47, pp. 88–107, Sep. 2013.

[3] Z. Yuan, G. Olsson, R. Cardell-Oliver, K. van Schagen, A. Marchi, A. Deletic, C. Urich, W. Rauch, Y. Liu, and G. Jiang, ''Sweating the assets—The role of instrumentation, control and automation in urban water systems,'' *Water Res.*, vol. 155, pp. 381–402, May 2019.

[4] Y. Qiu, Y. Liu, and D. Huang, ''Date-driven soft-sensor design for biological wastewater treatment using deep neural networks and genetic algorithms,'' *J. Chem. Eng. Jpn.*, vol. 49, no. 10, pp. 925–936, 2016.

[5] G. Boyd, D. Na, Z. Li, S. Snowling, Q. Zhang, and P. Zhou, ''Influent forecasting for wastewater treatment plants in North America,'' *Sustainability*, vol. 11, no. 6, p. 1764, Mar. 2019.

[6] B. Salman and O. Salem, ''Modeling failure of wastewater collection lines using various section-level regression models,'' *J. Infrastruct. Syst.*, vol. 18, no. 2, pp. 146–154, Jun. 2012.

[7] X. Yuan, Z. Ge, B. Huang, Z. Song, and Y. Wang, ''Semisupervised JITL framework for nonlinear industrial soft sensing based on locally semisupervised weighted PCR,'' *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 532–541, Apr. 2017.

[8] D. S. Lee, M. W. Lee, S. H. Woo, Y.-J. Kim, and J. M. Park, ''Nonlinear dynamic partial least squares modeling of a full-scale biological wastewater treatment plant,'' *Process Biochem.*, vol. 41, no. 9, pp. 2050–2057, 2006.

[9] B. Suchetana, B. Rajagopalan, and J. Silverstein, ''Investigating regime shifts and the factors controlling total inorganic nitrogen concentrations in treated wastewater using non-homogeneous hidden Markov and multinomial logistic regression models,'' *Sci. Total Environ.*, vol. 646, pp. 625–633, Jan. 2019.

[10] H. Guo, K. Jeong, J. Lim, J. Jo, Y. M. Kim, J.-P. Park, J. H. Kim, and K. H. Cho, ''Prediction of effluent concentration in a wastewater treatment plant using machine learning models,'' *J. Environ. Sci.*, vol. 32, pp. 90–101, Jun. 2015.

[11] P. Zhou, Z. Li, S. Snowling, B. W. Baetz, D. Na, and G. Boyd, ''A random forest model for inflow prediction at wastewater treatment plants,'' *Stochastic Environ. Res. Risk Assessment*, vol. 33, no. 10, pp. 1781–1792, Oct. 2019.

[12] M. Bongards, ''Improving the efficiency of a wastewater treatment plant by fuzzy control and neural networks,'' *Water Sci. Technol.*, vol. 43, no. 11, pp. 189–196, Jun. 2001.

[13] M. Zeinolabedini and M. Najafzadeh, ''Comparative study of different wavelet-based neural network models to predict sewage sludge quantity in wastewater treatment plant,'' *Environ. Monitor. Assessment*, vol. 191, no. 3, pp. 1–25, Mar. 2019.

[14] H. Liu, M. Huang, and C. Yoo, ''A fuzzy neural network-based soft sensor for modeling nutrient removal mechanism in a full-scale wastewater treatment system,'' *Desalination Water Treatment*, vol. 51, nos. 31–33, pp. 6184–6193, Sep. 2013.

[15] B. Ráduly, K. V. Gernaey, A. G. Capodaglio, P. S. Mikkelsen, and M. Henze, ''Artificial neural networks for rapid WWTP performance evaluation: Methodology and case study,'' *Environ. Model. Softw.*, vol. 22, no. 8, pp. 1208–1216, Aug. 2007.

[16] F. S. Mjalli, S. Al-Asheh, and H. E. Alfadala, ''Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance,'' *J. Environ. Manage.*, vol. 83, no. 3, pp. 329–338, May 2007.

[17] I. Bisschops, H. Spanjers, and K. Keesman, ''Automatic detection of exogenous respiration end-point using artificial neural network,'' *Water Sci. Technol.*, vol. 53, nos. 4–5, pp. 273–281, Feb. 2006.

[18] H. Poutiainen, H. Niska, H. Heinonen-Tanski, and M. Kolehmainen, ''Use of sewer on-line total solids data in wastewater treatment plant modelling,'' *Water Sci. Technol.*, vol. 62, no. 4, pp. 743–750, Aug. 2010.

[19] P. Antwi, J. Li, J. Meng, K. Deng, F. K. Quashie, J. Li, and P. O. Boadi, ''Feedforward neural network model estimating pollutant removal process within mesophilic upflow anaerobic sludge blanket bioreactor treating industrial starch processing wastewater,'' *Bioresource Technol.*, vol. 257, pp. 102–112, Jun. 2018.

[20] H. Chen, A. Chen, L. Xu, H. Xie, H. Qiao, Q. Lin, and K. Cai, ''A deep learning CNN architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources,'' *Agricult. Water Manage.*, vol. 240, Oct. 2020, Art. no. 106303.

[21] I. Pisa, I. Santin, A. Morell, J. L. Vicario, and R. Vilanova, ''LSTM-based wastewater treatment plants operation strategies for effluent quality improvement,'' *IEEE Access*, vol. 7, pp. 159773–159786, 2019.

[22] I. Pisa, I. Santín, J. L. Vicario, A. Morell, and R. Vilanova, ''A recurrent neural network for wastewater treatment plant effuents' prediction,'' in *Proc. 39th Jornadas de Automática*. Badajoz, Spain: Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura, 2018, pp. 621–628.

[23] T. Cheng, F. Harrou, F. Kadri, Y. Sun, and T. Leiknes, ''Forecasting of wastewater treatment plant key features using deep learning-based models: A case study,'' *IEEE Access*, vol. 8, pp. 184475–184485, 2020.

[24] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, ''On the properties of neural machine translation: Encoder-decoder approaches,'' 2014, *arXiv:1409.1259*.

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and D. L. Jackel, ''Backpropagation applied to handwritten zip code recognition,'' *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

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

[27] H. Hauduc, L. Rieger, A. Oehmen, M. C. M. van Loosdrecht, Y. Comeau, A. Héduit, P. A. Vanrolleghem, and S. Gillot, ''Critical review of activated sludge modeling: State of process knowledge, modeling concepts, and limitations,'' *Biotechnol. Bioeng.*, vol. 110, no. 1, pp. 24–46, Jan. 2013.

[28] A. Asadi, A. Verma, K. Yang, and B. Mejabi, ''Wastewater treatment aeration process optimization: A data mining approach,'' *J. Environ. Manage.*, vol. 203, pp. 630–639, Dec. 2017.

[29] P. Kadlec, B. Gabrys, and S. Strandt, ''Data-driven soft sensors in the process industry,'' *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, 2009.

[30] I. Jolliffe, ''Principal component analysis,'' in *Encyclopedia of Statistics in Behavioral Science*. Malden, MI, USA: Wiley, 2005.

[31] S. Wold, M. Sjöström, and L. Eriksson, ''PLS-regression: A basic tool of chemometrics,'' *Chemometrics Intell. Lab. Syst.*, vol. 58, no. 2, pp. 109–130, 2001.

[32] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1999.

[33] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[34] P. Chang and Z. Li, ''Over-complete deep recurrent neutral network based on wastewater treatment process soft sensor application,'' *Appl. Soft Comput.*, vol. 105, Jul. 2021, Art. no. 107227.

[35] M. M. Hamed, M. G. Khalafallah, and E. A. Hassanien, ''Prediction of wastewater treatment plant performance using artificial neural networks,'' *Environ. Model. Softw.*, vol. 19, no. 10, pp. 919–928, Oct. 2004.

[36] L. Belanche, J. J. Valdés, J. Comas, I. R. Roda, and M. Poch, ''Prediction of the bulking phenomenon in wastewater treatment plants,'' *Artif. Intell. Eng.*, vol. 14, no. 4, pp. 307–317, Oct. 2000.

[37] X. Yuan, L. Li, and Y. Wang, ''Nonlinear dynamic soft sensor modeling with supervised long short-term memory network,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3168–3176, May 2019.

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning internal representations by error propagation,'' California Univ. San Diego La Jolla Inst. Cogn. Sci., San Deigo, CA, USA, Tech. Rep. ICS Report, 8506, 1985.

[39] J. Qiao, X. Huang, and H. Han, ''Recurrent neural network-based control for wastewater treatment process,'' in *Proc. Int. Symp. Neural Netw.* Berlin, Germany: Springer, 2012, pp. 496–506.

[40] N. V. Bhattacharjee and E. W. Tollner, ''Improving management of windrow composting systems by modeling runoff water quality dynamics using recurrent neural network,'' *Ecol. Model.*, vol. 339, pp. 68–76, Nov. 2016.

[41] B. Mamandipoor, M. Majd, S. Sheikhalishahi, C. Modena, and V. Osmani, ''Monitoring and detecting faults in wastewater treatment plants using deep learning,'' *Environ. Monitor. Assessment*, vol. 192, no. 2, pp. 1–12, Feb. 2020.

[42] H. Kang, S. Yang, J. Huang, and J. Oh, ''Time series prediction of wastewater flow rate by bidirectional LSTM deep learning,'' *Int. J. Control, Autom. Syst.*, vol. 18, no. 12, pp. 3023–3030, Dec. 2020.

[43] G. Seo, S. Yoon, M. Kim, C. Mun, and E. Hwang, ''Deep reinforcement learning-based smart joint control scheme for on/off pumping systems in wastewater treatment plants,'' *IEEE Access*, vol. 9, pp. 95360–95371, 2021.

[44] S. Hochreiter, ''The vanishing gradient problem during learning recurrent neural nets and problem solutions,'' *Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

[45] B. Ryabko, ''Compression-based methods for nonparametric prediction and estimation of some characteristics of time series,'' *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 4309–4315, Sep. 2009.

[46] X. Du, Y. Cai, S. Wang, and L. Zhang, ''Overview of deep learning,'' in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Automat. (YAC)*, 2016, pp. 159–164.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[48] H. Yang, J. Li, and F. Ding, ''A neural network learning algorithm of chemical process modeling based on the extended Kalman filter,'' *Neurocomputing*, vol. 70, nos. 4–6, pp. 625–632, Jan. 2007.

[49] G. Doquire and M. Verleysen, ''Feature selection with missing data using mutual information estimators,'' *Neurocomputing*, vol. 90, pp. 3–11, Aug. 2012.

[50] C. Yan and Y. Zang, ''DPARSF: A MATLAB toolbox for 'pipeline' data analysis of resting-state fMRI,'' *Frontiers Syst. Neurosci.*, vol. 4, p. 13, May 2010.

[51] A. Botchkarev, “Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology,” 2018, *arXiv:1809.03006*.

[52] F. Chollet, “Keras,” Astrophysics Source Code Library, 2015. [Online]. Available: https://keras.io

[53] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[55] T. Fushiki, “Estimation of prediction error by using K-fold cross-validation,” *Statist. Comput.*, vol. 21, no. 2, pp. 137–146, Apr. 2011.

[56] B. Lim and S. Zohren, “Time-series forecasting with deep learning: A survey,” *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 379, no. 2194, Apr. 2021, Art. no. 20200209.

[57] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[58] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[59] H. Asami, M. Golabi, and M. Albaji, “Simulation of the biochemical and chemical oxygen demand and total suspended solids in wastewater treatment plants: Data-mining approach,” *J. Cleaner Prod.*, vol. 296, May 2021, Art. no. 126533.

[60] V. M. Kaya, J. de la Noüe, and G. Picard, “A comparative study of four systems for tertiary wastewater treatment by scenedesmus bicellularis: New technology for immobilization,” *J. Appl. Phycol.*, vol. 7, no. 1, pp. 85–95, Feb. 1995.

[61] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

MAIRA ALVI is currently pursuing a Ph.D. degree with The University of Western Australia (UWA). Her research interests include sensor data analytic using data mining techniques, machine learning, and its applications for real world complex industrial processes. She is a recipient of the prestigious Scholarship for International Research Fees (SIRF).

TIM FRENCH received the Honors degree in pure mathematics and the Ph.D. degree in computer science from The University of Western Australia. He is currently working as a Senior Lecturer in computer science and software engineering with The University of Western Australia. He is a Researcher primarily working in the fields of logic, artificial intelligence, and knowledge representation and reasoning. He is also interested in applying these ideas in industrial settings. He has worked on projects using automated planning for industrial process, capturing tacit and implicit knowledge from domain experts, and machine learning for complex processes.

RACHEL CARDELL-OLIVER received the master’s degree in computer science from The University of Western Australia (UWA) and the Ph.D. degree in formal methods for distributed systems from the University of Cambridge. She has worked at the University of Essex, U.K., and UWA, where she is currently the Head of Department of Computer Science and Software Engineering. She studies distributed sensor networks and designing systems that integrate data measurement using environmental sensors, data collection with wireless communication systems, and data analysis using data mining techniques. Working with multi-disciplinary teams, she has researched environmental challenges, such as understanding public transport use, reducing household water consumption, measuring water use by native Australian plants, and the performance of rammed earth for sustainable buildings in outback Australia.

PHILIP KEYMER received the Ph.D. degree from the School of Chemical Engineering, The University of Queensland (UQ). His thesis focused on the optimization of algal production with a novel electrochemical system and the anaerobic digestion of algal biomass grown on effluent water. He has industrial experience working for and with large utility companies and environmental service companies focusing on algal water treatment technology development and implementation. He is currently a Postdoctoral Research Fellow at the Australian Centre for Water and Environmental Biotechnology, UQ. In his current role, he is overseeing the delivery of two demonstration systems showcasing the integrated high rate anaerobic and high rate algae wastewater treatment process in collaboration with several large utility companies.

ANDREW WARD is currently employed as an Advance Queensland Industry Research Fellow at the Australian Centre for Water and Environmental Biotechnology (ACWEB), The University of Queensland. In this position, he works on several water related pilot scale projects, including waste water remediation utilizing microalgae, Anammox, and nutrient recovery utilizing pilot scale electrodialysis methods. He is also the Lead Investigator and a Project Manager for industry partner, Urban Utilities Microalgae Research Program.