Water Supply

Water Supply Vol 22 No 1, 424 doi: 10.2166/ws.2021.261

A comprehensive comparative analysis of deep learning tools for modeling failures in smart water taps

N. M. Offiong [a,*], Y. Wu[b], D. Muniandy[c] and F. A. Memon [a]

[a] University of Exeter, Centre for Water Systems, University of Exeter, Exeter, EX4 4QF, UK
[b] Department of Computer Science, EMPS, University of Exeter, Exeter, EX4 4QF, UK
[c] Imperial College London, Exhibition Rd, South Kensington, London SW7 2BU, UK
*Corresponding author. E-mail: no270@exeter.ac.uk

NMO, 0000-0002-0233-398X

ABSTRACT

Predicting early-stage failure in smart water taps (SWT) and selecting the most efficient tools to build failure prediction models are many challenges that water institutions face. In this study, three Deep Learning (DL) algorithms, i.e., the Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (BiLSTM), were selected to analyse and determine the most appropriate among them for failure prediction in SWTs. This study uses a historical dataset acquired from smart water withdrawal taps to determine the most efficient DL neural network architecture for failure prediction in the SWT, leading to a hybrid model's development. After a comprehensive evaluation of the three ML models, findings show that a hybrid combination of the CNN and Bi-LSTM (CNN-BiLSTM) models is a better solution for investigating failures in the SWT.

Key words: Bi-LSTM, CNN, deep learning, failure prediction, LSTM, smart water taps, time-series

HIGHLIGHTS

- Automated feature extraction from smart tap time series dataset.
- The development of a temporal dependent model to explore time series dataset.
- A comprehensive analysis of three applied deep learning models.
- The development of a proposed hybrid model (CNN-BiLSTM) for failure prediction in SWTs deployed in rural Africa.
- Model evaluation with real world dataset.

INTRODUCTION

Early detection and prediction of smart water taps (SWT) failures are challenges affecting domestic water services delivery in rural Africa. Water service providers have invested significant time and financial resources in the sustainability of water supply and withdrawal systems in most parts of Africa for daily domestic water consumption (Dungumaro 2007; Aberilla *et al.* 2020). However, constant fault development and outright failure of some water withdrawal taps have been a daunting problem for both the water users and the service providers (Smiley 2013; Hughes 2019). In effect, this problem gives rise to an unsteady water supply (Burt *et al.* 2018). To this end, the need to detect the occasional failures in the water systems before they occur deserves serious attention to initiate pro-active predictive maintenance programmes. A potential way to address this problem is by harnessing the data acquired from the SWTs as they represent the taps' behaviour.

Daily water usage generates useful time-series datasets that carry information about the variables that determine the SWT system's dynamics. Analysing the dataset using the most appropriate tools and techniques will help proffer meaningful solutions to water supply failures by predicting the SWT tap's failures. However, the success of a model or a method for time-series data analysis is predominantly affected by underlying factors such as the kind of data and the approach taken to analyse the data (Brownlee 2020).

Conventionally, there are different models and methods for time series forecasting. These methods create uncertainties when incorrectly applied to specific case studies and can lead to wrong interpretations. Wei *et al.* (2020) applied three

different ML networks to analyse time-series data from pore-water; the study investigated the Gated Recurrent Units (GRU), LSTM and standard Recurrent Neural Networks (RNN). The study showed that the LSTM and GRU models performed better on the data. Xu *et al.* (2020) applied a deep learning model based on LSTM to predict a water supply network. The study revealed that the LSTM model effectively handles the shortcomings of ordinary artificial neural networks when predicting complex conditions in Smart Water Networks. In their work, Muharemi *et al.* (2018) presented some approaches for abnormal event detection on aquatic time series data. Chen *et al.* (2020) found that a deep learning method based on neural networks outperforms the support vector machine (SVM). The authors reported that their models could be trained and learned automatically for prediction purposes given different data samples. Each ML predictive modelling project is different but shares the necessary implementation steps.

In this research, we consider the nature of the data and our expectations from the analysis. Notably, the data used for this analysis is a real time-series dataset with some missing information (noise). When tabulated, the data contains columns with single values that need to be identified and removed for efficient data cleaning. Also, the data is such that it contains column variables with few unique values and duplicate rows. These inconsistencies require proper identification and correction where appropriate. As such, we explore DL algorithms that are robust and can handle datasets with such discrepancies. These include (i) CNN (ii) LSTM and (iii) Bi-LSTM.

We base the selection of CNN for our study on its ability to resist noisy data and automatically extract in-depth local features from a dataset with minimal preprocessing irrespective of the dataset's discrepancies (Wang *et al.* 2018). CNN can also discover and extract the relevant deep structures to reveal the input time series' in-depth features using convolution and pooling operations. The choice of LSTM comes from its ability to classify temporal features over long sequences of data, taking into consideration the salient features of the dataset (Xu *et al.* 2020). Similarly, the selection of Bi-LSTM hinges on its ability to improve the LSTM model's performance by running inputs in two directions – past and future (Graves *et al.* 2005; Yin *et al.* 2020). Apart from these three DL models, other traditional models for time series analysis do not have the *deep* capability of the three models selected for this study. These traditional models include: ARIMA (Siami-Namini *et al.* 2019), Autoregressive (AR) model (Chen & Wang 2019), and Support Vector Machine (SVM) (Mamun *et al.* 2020).

The main aim of this research centres on applied machine learning techniques, where we apply three different DL techniques to make predictions on the structural and functional features of smart water withdrawal taps through the use of the data generated by SWTs. A comparative analysis of the DL tools was done to ascertain the three models' efficiency concerning the accuracy, F1 score, precision and recall measures. Individually, the LSTM, Bi-LSTM and CNN models have shown promising results in their respective capacities for real-world time series problems (Yu *et al.* 2017; Muñoz *et al.* 2020; Yin *et al.* 2020). However, to have a single model capable of combining the automatic feature extraction of the CNN model and simultaneously having the memory retention of the LSTM model, we combined two of the models to build a hybrid model for failure prediction in SWTs. The proposed hybrid model was built from a fusion of two of the most promising three study models – the CNN and Bi-LSTM (CNN-BiLSTM).

The main contributions of this paper include:

- Automated feature extraction from time series dataset acquired from a set of SWTs deployed in rural Africa without excessive hyperparameter tuning.
- The development of a temporal dependent model to explore heterogeneous noisy time series datasets.
- A comprehensive analysis of three applied deep learning models based on accuracy, F1 Score, precision and recall measures, time complexity, parallelisation, and bias-variance trade-off.
- A proposed hybrid model (CNN-BiLSTM) was developed for failure prediction in SWTs deployed in rural Africa. This research is the first time this kind of hybrid model is applied to study solar-powered smart water withdrawal taps.

METHODS

Data and preprocessing

The DL process of failure prediction requires that we split our dataset into two main parts, namely: (i) the training set (including a validating dataset) and (ii) the testing set. The training set consists of the functional (normal water tap design flow) and nonfunctional (failure of the water tap) samples of the water taps in the study area (García *et al.* 2016). On the other hand, the testing set is part of the original dataset put aside during the data split and used to assess the classifier's efficiency. The testing set can also be used to obtain a separate evaluation of the test hypothesis (Ripley 1996). The data used for this study consist of

a record with 1,047,114 rows representing time-series samples acquired from 27 different SWTs installed in a rural part of a sub-Saharan African village. It has 22 columns describing the features of the dataset. The time-series data samples used for our study consist of some missing information and similar sequences. These discrepancies can result in increased time for the DL algorithm implementation because of the noise in prediction.

This research carried out extensive data preparation, which exposes most of the problem's underlying structure to the DL algorithms. We began by carrying out data cleaning, identifying and correcting errors/mistakes in the dataset. We then carried out feature selection by identifying input variables that are relevant to our study. Following the two phases of data preparation highlighted above, we engaged in data transformation, changing some of the data distribution. Since our data is a numeric dataset with accompanying subtypes, we applied normalisation transform by rescaling the variables to a range of 0 and 1 to accommodate all the dataset values. We used the *Min-Max* method to transform the variables' values to decimal values between 0 and 1 and consider their minimum and maximum values (Thara *et al.* 2019). The formula given in Equation (1) shows the Min-Max normalisation:

$$x_{(0,1)} = \frac{x - Min}{Max - Min} \tag{1}$$

where $x_{(0,1)}$ and x are the normalised and original datasets, respectively. *Min* and *Max* show the minimum and maximum values in the entire dataset, respectively. After normalising the dataset, we engaged in feature engineering by deriving a new variable from the original dataset. Then we created a compact projection of the dataset by reducing dimensionality.

Attribute extraction and selection

We establish that different attributes of the SWTs play significant roles in the data collected from the water supply taps. However, not all the attributes of the dataset are relevant for our failure prediction analysis. Also, copious and redundant attributes can lead to the 'curse of dimensionality,' limiting the model's generalisation and increasing the model run time. Therefore, the proposed research ignores the less relevant data attributes and combines some other attributes of similar data. Table 1, shown below, gives a representation of attributes extracted and selected from the dataset.

Observe also that there were initially twenty-two columns and attributes. Having irrelevant features could have potentially affected the ML algorithms' implementation, increased implementation time, and slowed down the model. The twenty-two columns and attributes were cut down to just five relevant columns and attributes to reduce the redundant and irrelevant attributes on the ML algorithms during implementation. The process of reducing the 22 columns and attributes to five is based on logical reasoning, mainly because the dataset contains features or columns that do not affect the model training and have no effect on the proposed model's outcome. The selected five columns can clearly explain our aims of building this model and do not include any redundant attributes in the final dataset. After extracting and selecting relevant features from the SWTs, the classification algorithms were tested separately with weighted attributes in increasing order to determine the one with the most efficient promises for SWTs.

Main development tools used

The development environment was built as follows:

I. Anaconda distribution version – Conda 4.8.3
II. Major development libraries used – Keras 2.1.0 (with Tensorflow backend), Livelossplot 0.5.3, Scikit-learn 0.23.2
III. Cloud platform used to do model training – Google Colab Pro

Table 1 | Attribute extraction and selection

Column header	Column description
AssetID	Unique identifier for each SWT (Tap number)
ErrorCode	Codes indicating issues on water tap usage
FlowRate	The volume of water collected over time
DateTime	Time and date of water withdrawal
Voltage	The battery voltage capacity at each instance the water tap accessed

The dataset consists of 28 different water taps (assets). The proposed deep learning solution was implemented on each of these water taps (assets). We solved two kinds of prediction problems in this study, a classification challenge (for the Error code), the other being a regression challenge (for the Flow rate). We have experimented with three different DL architectures in each of the classification and regression challenges:

(a) LSTM
(b) BiLSTM
(c) CNN

After experimenting with the three DL models above, we realised that with a combination of the CNN and the Bi-LSTM models, we got a more robust algorithm that captures the dataset's salient features and keeps reliable memory of the learning process. That eventually led to the design of a hybrid CNN-BiLSTM model.

Each of the neural network architecture's final output results provided different accuracies, losses and other valuable metrics that helped evaluate the best performing neural network overall for our study. The aim is to reveal one or a hybrid neural network architecture to classify labels better and predict continuous values for our given water tap data.

Assumptions

The original dataset contains inconsistent time series. For example, in Figure 1 below, asset (tap) 5 shows some data inconsistencies.

The time does not seem to indicate if it is increasing per minute or hour. Similarly, we also observe inconsistencies when there is a change in dates (dates progressing to the next day). In summary, the time element is not consistent. Hence, we had to assume that each time progresses in an hourly manner. This challenge could be avoided if the real dataset was of higher quality.

Predicting error code (classification)

The classification solution was formulated to predict a given number of next multi-steps of Error code. This prediction helps in determining when water tap failures are bound to happen. If the dataset is measured in hourly time-steps, the prediction also produces output time-steps hourly.

The case study data was split into three sets: training, testing, and validation sets. The training data was used in the model building process. In contrast, the validation data, a smaller sub-split from the training data, was used to validate the model's performance – as the model is being built (in near real-time). Finally, as mentioned above, the testing data is an independent dataset not used in training or validation; and is used exclusively in testing the model's generalizability. The target column (Error code) must be converted to a categorical format (as a preprocessing step) before predictions.

After the categorical conversion, a time series splitting function helps compose the dataset into a supervised time-series format. The dataset was transformed into a 3-dimensional dataset that consists of [*rows*, *time-steps*, *features*]. Depending on the type of water tap used, the number of observation rows will differ. The time-splitting technique was applied to all the available data splits – training, validation, and testing.

Two other essential elements built into the solution are *Model check-pointing* and *Early stopping*. Model check-pointing helps monitor the F1 score for the validation dataset and saves the model that the neural network found to have the highest F1 score. Early stopping provides the architecture's flexibility to wait patiently for 50-number times before it completely stops

	DateTime
0	2004-01-01 01:35:09
1	2004-01-01 01:38:00
2	2004-01-01 01:39:02
3	2004-01-01 03:38:07
4	2004-01-01 03:39:09

Figure 1 | Data extract showing inconsistencies in the dataset.

training the model. Suppose no improvement was found in the F1 score (in this case, the F1 score should be increasing). Figures 2–5 below show the different model architectures used for the study.

As illustrated in Figure 2 above, Table 2 below shows the hyperparameters set for the LSTM architecture.

The Bi-LSTM architecture is shown in Figure 3 and illustrated in Table 3 below. It should also be observed that the output layer has the equivalent number of neurons as the multiple classes of *Errorcodes*.

Table 4 below shows the hyperparameters used in the development of the CNN model for our study.

At this point, we propose a CNN-BiLSTM, which is a hybrid combination of two DL models – the CNN and bidirectional LSTM. The motivation behind combining these two techniques is to harness the CNN model's automatic feature extraction capability and combine it with the long-term memory retention of the Bi-LSTM model. It also considers the temporal nature of the dataset from the SWTs. As shown in Figure 5 below and presented in Table 5, the CNN layer automatically extracts salient features from the time series dataset. It forms the front end of the combined architecture. At the front end, the spectral features X_i, $i = 1,2,\ldots,$ n are passed as input into the model. The CNN layers act as feature extractors and use the (kr $\times$ kc) convolutional kernel size to effectively extract the temporal features from the input data.

The Bi-LSTM, on the other hand, consists of two LSTM enabled bi-directional recurrent neural networks (BiRNN), which are standard neural networks (Byeon *et al.* 2015). The motivation behind using the Bi-LSTM model is that each training step has a forward and backward LSTM enabled RNN connected to an output layer. As such, each point keeps a memory of the

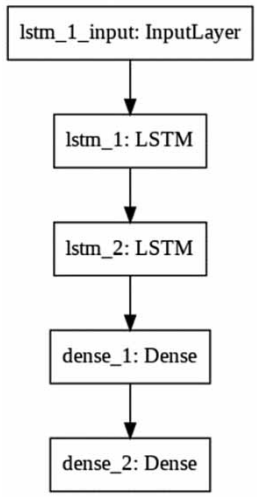

Figure 2 | LSTM neural network architecture.

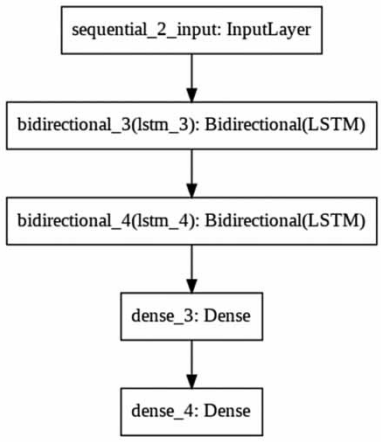

Figure 3 | Bi-LSTM neural network architecture.

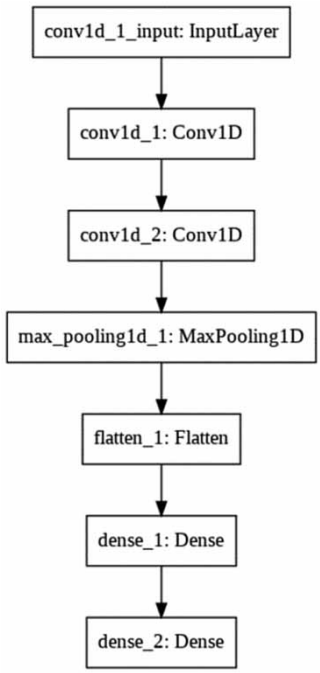

Figure 4 | CNN neural network architecture.

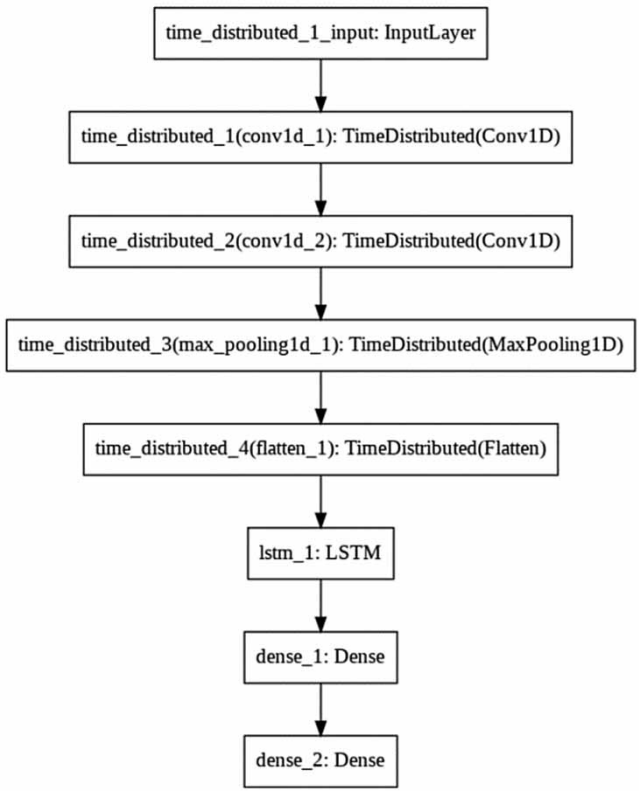

Figure 5 | CNN-BiLSTM neural network architecture.

Table 2 | LSTM hyperparameters

LSTM					
Input layer	**Hidden layer**	**Neuron**	**Dense layer**	**Dense output layer**	**Sequence length**
1	2	100	1	1	64
Params	564,879				

Table 3 | Bi-LSTM hyperparameters

BiLSTM					
Input layer	**Hidden BiLSTM layer**	**Neuron**	**Dense layer**	**Dense output layer**	**Sequence length**
1	2	100	1	1	64
Params	546,026				

Table 4 | CNN hyperparameter settings

CNN						
Conv layer	**Maxpooling layer**	**Neuron (in each layer)**	**Flattening layer**	**Dense layer**	**Filter**	**Output layer**
1	1	100	1	1	64	1
Params	349,025					

Table 5 | CNN-BiLSTM hyperparameter setting

CNN-BiLSTM								
Conv layer	**Maxpooling layer**	**Neuron (in each layer)**	**Flattening layer**	**Dense layer**	**Filter**	**BiLSTM layer**	**Dense layer**	**Dense output layer**
	1	100	1	1	64	1	1	1
Params	625,800							

past and future memory of the training. Thus, the Bi-LSTM component accommodates the dataset's temporal nature in the proposed model while keeping a long short-term memory of the model training.

As illustrated in Figure 5, Table 5 shows that the proposed CNN-BiLSTM model has 2 *Time distributed* layers, 1 *Max pooling* layer, 1 *Flattening layer*, 1 *BiLSTM layer*, and 1 Dense and 1 *Dense Output* layer. Each time, the distributed layer has 64 filters and three kernels. Thus, each layer has a constant number of 100 neurons in each hidden layer.

Predicting flow rate (regression)

We also formulate a regression solution to predict a given number of Flowrate's next hour (Mamun *et al.* 2020). Given the historical data values, predicting the next hour flow rate helps determine how the flow rates are likely to change. The previous classification example shows that the dataset is split into training, testing, and validation sets. Testing data uses the last 12 hourly time-steps from the original (unchanged) dataset. Thus, a higher testing data value will result in a longer training time and becomes computationally 'expensive' to execute.

The time series data must be converted to a supervised learning format, and a helper defined to generate this (*series_to_supervised*). The function allows the user to specify the number of lag observations to use in the input. Also, to specify the number to use in the output (*n_out*) for each data sample. It will also remove all rows containing NaN (Not a Number) values as they cannot train or test the model.

We used one of the most promising evaluation techniques to evaluate model performance on a testing set – the walk-forward model evaluation technique (Falessi *et al.* 2018). Hence, we have implemented this functionality in our solution. The walk-forward validation is an approach where the model makes a forecast for each observation in the testing dataset by taking the data one at a time. At the end of each forecast for a time step in the testing dataset, the forecast's accurate observation is added to the testing dataset and made available to the model. Simpler models can be refit with the observation before making subsequent predictions. As seen in our present solution, more sophisticated models (such as neural networks) do not refit each time, given the much higher computational cost. However, the exact observation for the time step can then predict the next step as a section of the input.

Due to the stochastic nature of neural networks and the possibility of obtaining different results each time, we have also implemented a '*repeat evaluate*' functionality within the solution itself. The *repeat_evaluate* helper function allows model training to occur many times, and an average of the runs is reported as the error value. Given the same model hyper-parameter configuration and the same training dataset, a different internal set of weights and biases could result in each time the model is trained, which has different evaluation results. The challenge arises when evaluating a model's performance and choosing a final model to make predictions. To counter this challenge, we have implemented a *repeat_evaluate* helper function that repeats model training (n) several times before reporting the average RMSE error value. Again, this can be computationally expensive if there is no suitable hardware to run the model training process.

Finally, the proposed model's performance can be summarised from these repeated runs. A summary statistical approach is used, where the mean and the standard deviation values are used to find the average values for each. One final mean and standard deviation value(s) are output as a result.

Model evaluation metrics

The evaluation metrics used in this research include *Accuracy, F1 Score, Precision and Recall.* These metrics help to evaluate the generalisation ability of the trained model classifier. In addition, we use the evaluation metrics to measure and summarise the classifier's quality after training the model. The accuracy (or error rate) metric is one of the most frequently used metrics in practice. Several researchers have employed it to evaluate the generalisation ability of DL classifiers (Mohammad & Sulaiman 2015; Mosavi *et al.* 2020). Through the accuracy metric, the trained classifier is measured based on the total instances correctly predicted by the trained classifier when tested with the new data. Equation (2) below shows the accuracy evaluation metric.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

where TP – true positive

TN – true negative
FP – false positive
FN – false negative

We also harnessed the F1 score to evaluate our model because the F1 score helps to measure both Recall and Precision metrics together. Hence, we have one metric value that simplifies the understanding of our results (Mosavi *et al.* 2021). Secondly, the F1 score provides a single 'harmonic mean' value to ensure that false positives and false negatives are considered together. Thus, this metric offers an excellent benefit (one metric summary). Equation (3) below shows the F1 score metric.

$$\text{F1Score} = \frac{2*(Recall*Precision)}{Recall + Precision} \tag{3}$$

Another evaluation metric considered in this research is the Precision metric. The precision metric gives a measure of correct positive predictions from the total prediction in the positive class. The formula is shown in Equation (4)

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

The Recall metric was also used to measure the fraction of positive patterns that our model correctly classified. The recall formula is shown in Equation (5)

$$\text{Recall} = \frac{TP}{TP + TN} \tag{5}$$

As mentioned above, the evaluation metrics allow us to select the optimal solution to build a robust predictive classifier.

RESULTS AND DISCUSSION

This research carried out a comparative analysis of failure prediction techniques in time series data using LSTM, CNN, BiLSTM and CNN-BiLSTM models. The models were built using simple principles and techniques, with strong consideration for the nature of the real dataset used for the study. The paper adopts the data-driven approach for failure prediction in smart water taps. The four models in the experiments used the same real dataset acquired from real solar-powered water taps installed in Sub Saharan Africa. The first task was to preprocess the datasets to remove impurities and used them as input for the prediction. The three models (i.e., the LSTM, BiLSTM and CNN models) chosen for this study and the proposed hybrid model (i.e., the CNN-BiLSTM model) can categorise and predict failures for a real dataset. In addition, they can further be adjusted to suit the demand of future time series.

As shown in Table 6, the first three models performed well during the experiment's respective capabilities. However, this study compares the three models' efficiency and develops a scalable model with a more robust application in our case study. The desire is to build a more scalable model with the ability to extract salient features from the SWT dataset automatically and, at the same time, be able to keep track of long dependencies during model training. The proposed model based on a hybrid CNN-BiLSTM architecture proved to be very robust in its accuracy. The proposed model also has a better overall performance compared to the other models used in the experiment.

Each water tap (asset) provides a near real-time training plot for the CNN-BiLSTM model, as shown in Figure 6(a)–6(e), taking tap number five as an example. After each epoch training (the number of passes completed by the ML algorithms), the plot is updated interactively. At the end of all epochs, we see the resulting final plot, which visualises all the classification-evaluation metrics, as shown in Figure 6(a)–6(e). Figure 6(a) shows the model's accuracy measure of 88%, Figure 6(b) model's F1 score of 0.87, Figure 6(c) shows the overall model's loss, which is 0.04, Figure 6(d) shows the model's precision of 0.81, while Figure 6(e) shows the model's Recall measure of 0.91.

The accuracy and loss metrics are visualised for training and validation datasets. However, there are only validation dataset visualisations for F1 Score, Precision and Recall scores. The reason is that we only need to consider the performance of the trained model against validation datasets to measure the fitness of our predictions. Our study concentrated on improving the F1 score because it is the best metric to evaluate our experiment. The F1 score was used as a yardstick for selecting the better model for our SWT time series dataset.

Among the different metrics considered in this research (Accuracy, F1 Score, Loss, Precision and Recall), the F1 score is considered a critical metric in the design of our proposed solution based on the following reasons:

A custom Keras metric is written to evaluate the F1 Score, Recall and Precision scores. This custom code is called inside the Keras callback function, and the evaluation is performed on the validation dataset; the evaluation is shown in Table 7 below. Each of the metrics has a total value computed for all epochs: *min* (minimum), *max* (maximum) and *cur* (current) values. These metrics help in deciding the best model from the list of models used for the experiment.

Table 6 | Models' performance evaluation

Model	Accuracy	F1 Score	Precision	Recall	Loss	Train Time (Minutes)
LSTM	0.824	0.880	0.880	0.890	2.232	55.30
Bi-LSTM	0.888	0.850	0.810	0.910	0.578	85.08
CNN	0.841	0.830	0.860	0.840	3.447	2.43
CNN-BiLSTM	0.881	0.870	0.810	0.910	0.042	2.96

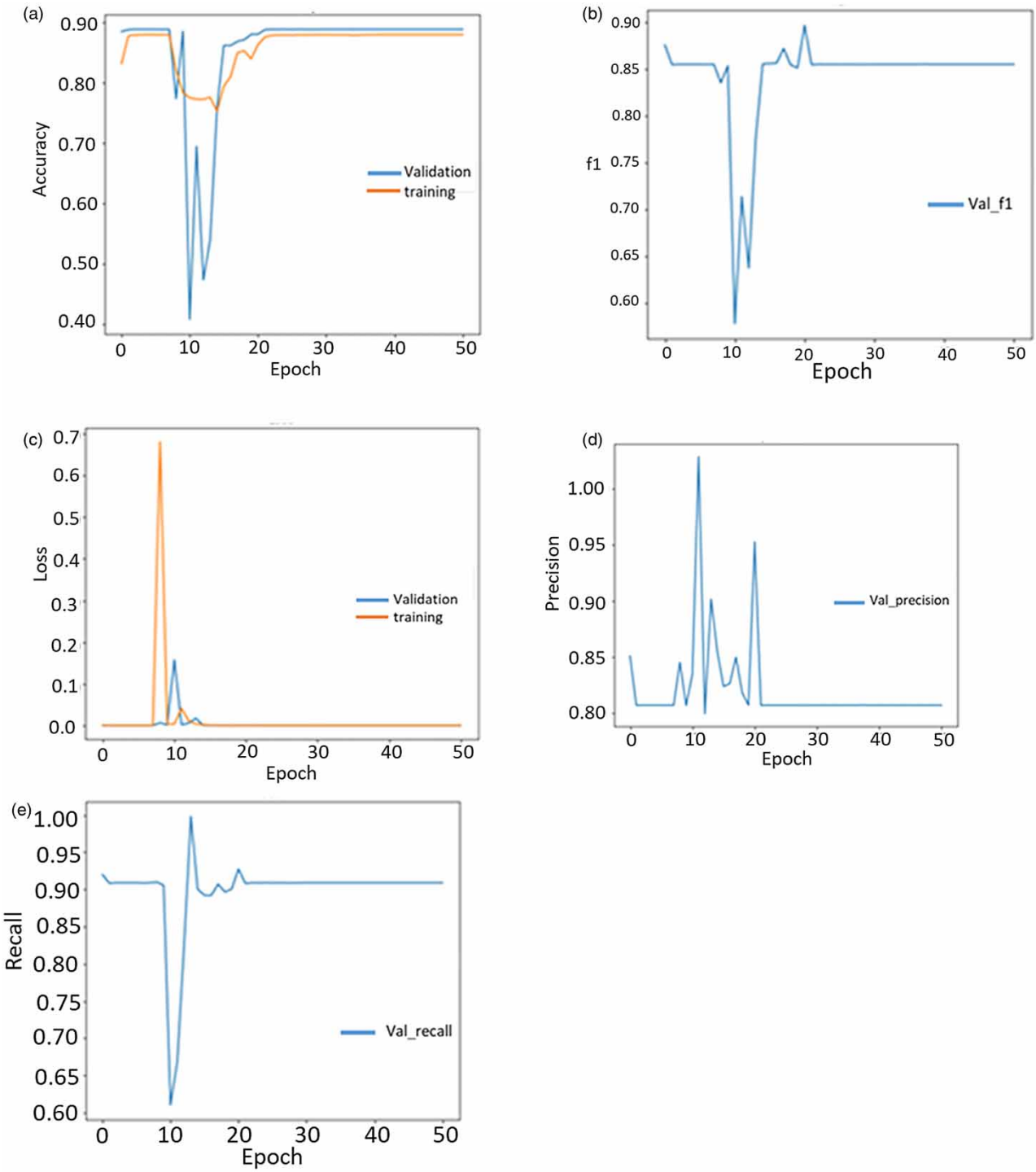

Figure 6 | (a–e): Model performance evaluation metrics (classification).

After the completion of model training, the model's classification performance is measured using the Area Under the Curve (AUC) and Receiver Operating Characteristics (ROC) metrics (Siddiqui *et al.* 2020). These metrics measure the performance of the classification model at different threshold settings. The AUC metric represents the degree of separability, while the ROC is a probability curve. Thus, the AUC-ROC tells how much the model is capable of distinguishing between classes. A point to note is that most of the data collected from the water taps are imbalanced; hence the *Macro* AUC-ROC does not

Table 7 | A sample of a tap (tap 5) evaluation results – classification

Accuracy							
	Validation	(min:	0.407,	max:	0.888,	cur:	0.888)
	Training	(min:	0.752,	max:	0.879,	cur:	0.879)
F1 score							
	val_f1	(min:	0.580,	max:	0.900,	cur:	0.870)
Loss							
	Validation	(min:	0.578,	max:	1,573.351,	cur:	0.578)
	Training	(min:	0.611,	max:	6,807.710,	cur:	0.612)
Precision							
	val_precision	(min:	0.800,	max:	0.100,	cur:	0.810)
Recall							
	val_recall	(min:	0.610,	max:	0.100,	cur:	0.910)

provide any values. This scenario is due to the missing classes found inside the actual ***y_test*** data split. The higher the AUC-ROC score, the better the model's classification.

Each water tap asset provides a near real-time training plot. For example, Figure 7(a) shows the proposed CNN-BiLSTM model's accuracy measure, while Figure 7(b) shows the hybrid model's MSE metric (based on regression). After each epoch training is completed, the plots are updated interactively. Thus, we can observe the resulting final plot at the end of all epochs that visualises all the regression-evaluation metrics – accuracy and mean squared error (MSE).

Again, as shown in Table 8 below, each of the metrics has a total value computed for all the epochs: *min* (minimum), *max* (maximum) and *cur* (current) values. There is also a total training time returned in minutes. There is also a final score of *20.660* in this example, and it is the MSE score for the trained model, measured in the original units for *Flowrate* – cubic meters per second (m^3/s). The results obtained further prove that the proposed CNN-BiLSTM is a better choice for failure prediction in SWTs.

This study compares three DL techniques, including the LSTM, BiLSTM and CNN models. Individually, these models have their various merits. For example, the CNN model can automatically extract our dataset's features without manually tuning the hyper-parameters but traditionally, the CNN model cannot retain memory. This shortcoming led to the introduction of the Bi-LSTM model, which can retain training information for longer sequences. Therefore, a hybrid fusion of these two

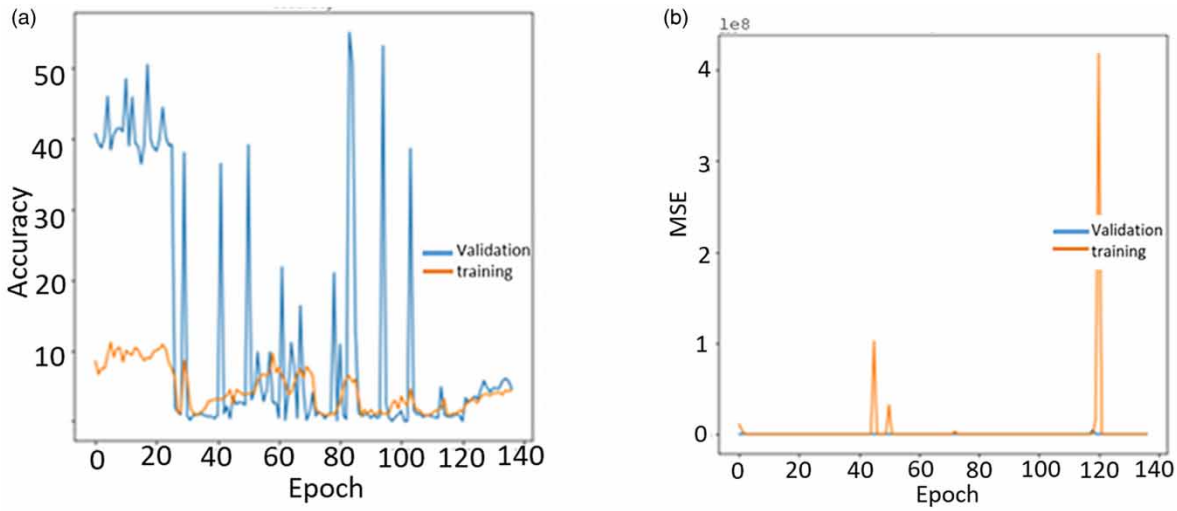

Figure 7 | (a, b): Model performance evaluation metrics (regression).

Table 8 | A sample of a tap (Tap 5) evaluation results – regression

Accuracy							
	Validation	(min:	0.300,	max:	0.550,	cur:	0.005)
	Training	(min:	0.090,	max:	0.110,	cur:	0.004)
MSE							
	Validation	(min:	0.00110,	max:	0.01500,	cur:	0.01500)
	Training	(min:	0.00180,	max:	0.04100,	cur:	0.04100)

most promising models for our study shows that the proposed BiLSTM-CNN model achieves very high accuracy and less processing time. Simultaneously, the proposed hybrid model can virtually explore the spatial-temporal information in the water tap dataset with an outstanding processing speed.

In this research, our interest was to compare three DL models for failure prediction and assess the extent to which these models can be used to predict failures in smart water taps deployed in sub-Saharan Africa. As Table 7 shows, it can be seen that all the models show promising results in their rights. However, a hybrid combination of the CNN and BiLSTM models gives a more promising result and captures the salient features of our dataset, which has noisy characteristics. Furthermore, as mentioned above, the choices of the CNN are to carry out automatic feature extraction, such that manual tuning of the hyperparameters will not be needed.

CONCLUSION

This research paper presented a comparative analysis of three different DL failure prediction techniques and compared them to a proposed hybrid model built from two selected techniques for failure prediction in SWTs deployed to rural Africa. The models investigated include the LSTM, CNN, BiLSTM and CNN-BiLSTM (the hybrid model) models. The comparison considered some vital evaluation metrics, including the Accuracy, F1 Score, Precision and Recall measures of the techniques, including the hybrid model. In addition, some crucial input information from the solar-powered electronic taps, including the flow rate and the error of the e-tap, were used to evaluate the three techniques and the hybrid model. The dataset used for the research was a real historical dataset with some inconsistencies as noises.

Nevertheless, we showed that all three DL techniques could discover latent information from the time series dataset and make informed predictions based on the predictions' results from the experiments' results. However, the proposed hybrid model, a combination of CNN and Bi-LSTM, appears to be the most effective for developing an early warning system for SWT deployed in rural Africa. Therefore, the CNN-BiLSTM will be built into an incremental model to work online in real-time for future studies.

ACKNOWLEDGEMENTS

This research is part of a PhD scholarship provided by the TETFund Scholarship Grant (Nigeria), thankfully acknowledged. The authors would also like to thank Rob Hygate, Roger Godwin, and other staff members of eWaterPay for making access to the e-Tap data available.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

REFERENCE

Aberilla, J. M., Gallego-Schmid, A., Stamford, L. & Azapagic, A. 2020 Environmental assessment of domestic water supply options for remote communities. *Water Research* **175**, 115687. https://doi.org/10.1016/j.watres.2020.115687.

Brownlee, J. 2020 *Data Preparation for Machine Learning*, v1.1 (Brownlee, J. ed.). Machine Learning Mastery, Australia.

Burt, Z., Ercümen, A., Billava, N. & Ray, I. 2018 From intermittent to continuous service: costs, benefits, equity and sustainability of water system reforms in Hubli-Dharwad, India. *World Development* **109**, 121–133. https://doi.org/10.1016/j.worlddev.2018.04.011.

Byeon, W., Breuel, T. M., Raue, F. & Liwicki, M. 2015 Scene labeling with LSTM recurrent neural networks. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07–12 June, pp. 3547–3555. https://doi.org/10.1109/CVPR.2015.7298977.

Chen, K., Chen, H., Zhou, C., Huang, Y., Qi, X., Shen, R., Liu, F., Zuo, M., Zou, M., Wang, J., Zhang, Y., Chen, D., Chen, X., Deng, Y. & Ren, H. 2020 Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data. *Water Research* **171**, 115454. https://doi.org/10.1016/j.watres.2019.115454.

Chen, Y. & Wang, K. 2019 Prediction of satellite time series data based on long short term memory-autoregressive integrated moving average model (LSTM-ARIMA). In: *2019 IEEE 4th International Conference on Signal and Image Processing, ICSIP 2019*, pp. 308–312. https://doi.org/10.1109/SIPROCESS.2019.8868350.

Dungumaro, E. W. 2007 Socioeconomic differentials and availability of domestic water in South Africa. *Physics and Chemistry of the Earth* **32** (15–18), 1141–1147. https://doi.org/10.1016/j.pce.2007.07.006.

Falessi, D., Narayana, L., Fong, J. & Turhan, B. 2018 *Preserving Order of Data When Validating Defect Prediction Models*, pp. 1–20.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M. & Herrera, F. 2016 Big data preprocessing: methods and prospects. *Big Data Analytics* **1** (1), 1–22. https://doi.org/10.1186/s41044-016-0014-0.

Graves, A., Fernández, S. & Schmidhuber, J. 2005 Bidirectional LSTM networks for improved phoneme classification and recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **3697** LNCS(May), 799–804.

Hughes, D. A. 2019 Facing a future water resources management crisis in sub-Saharan Africa. *Journal of Hydrology: Regional Studies* **23**, 100600. https://doi.org/10.1016/j.ejrh.2019.100600.

Mamun, M., Kim, J. J., Alam, M. A. & An, K. G. 2020 Prediction of algal chlorophyll-a and water clarity in monsoon-region reservoir using machine learning approaches. *Water (Switzerland)* **12** (1), 1–3. https://doi.org/10.3390/w12010030.

Mohammad, H. & Sulaiman, M. N. 2015 A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process* **5** (2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201.

Mosavi, A., Sajedi-Hosseini, F., Choubin, B., Taromideh, F., Rahi, G. & Dineva, A. A. 2020 Susceptibility mapping of soil water erosion using machine learning models. *Water (Switzerland)* **12** (7), 1–17. https://doi.org/10.3390/w12071995.

Mosavi, A., Sajedi Hosseini, F., Choubin, B., Goodarzi, M., Dineva, A. A. & Rafiei Sardooi, E. 2021 Ensemble boosting and bagging based machine learning models for groundwater potential prediction. *Water Resources Management* **35** (1), 23–37. https://doi.org/10.1007/s11269-020-02704-3.

Muharemi, F., Logofătu, D., Leon, F., Geetha, S., Gouthami, S., Lapikas, T. & Rieck, K. 2018 Machine learning approaches for anomaly detection of water quality on a real-world data set. *Journal of Information and Telecommunication* **2**, 1–14. https://doi.org/10.2166/hydro.2015.113.

Muñoz, D., Narváez, C., Cobos, C., Mendoza, M. & Herrera, F. 2020 Knowledge-based systems incremental learning model inspired in Rehearsal for deep convolutional networks. *Knowledge-Based Systems* **208**, 106460. https://doi.org/10.1016/j.knosys.2020.106460.

Ripley, B. D. 1996 *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

Siami-Namini, S., Tavakoli, N. & Siami Namin, A. 2019 A Comparison of ARIMA and LSTM in Forecasting Time Series. In: *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Siddiqui, M. K., Morales-Menendez, R. & Ahmad, S. 2020 Application of receiver operating characteristics (ROC) on the prediction of obesity. *Brazilian Archives of Biology and Technology* **63**, 190736. https://doi.org/10.1590/1678-4324-2020190736.

Smiley, S. L. 2013 Complexities of water access in Dar es Salaam, Tanzania. *Applied Geography* **41**, 132–138. https://doi.org/10.1016/j.apgeog.2013.03.019.

Thara, T. D. K., Prema, P. S. & Xiong, F. 2019 Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. *Pattern Recognition Letters* **128**, 544–550. https://doi.org/10.1016/j.patrec.2019.10.029.

Wang, Q., Li, H., Chen, Z., Member, S., Zhao, D., Ye, S. & Cai, J. 2018 Supervised and Semi-Supervised Deep Neural Networks for CSI-Based Authentication. *ArXiv:1807.09469 [Cs.LG]*.

Wei, X., Zhang, L., Yang, H. Q., Zhang, L. & Yao, Y. P. 2020 Machine learning for pore-water pressure time-series prediction: application of recurrent neural networks. *Geoscience Frontiers*. https://doi.org/10.1016/j.gsf.2020.04.011.

Xu, Z., Ying, Z., Li, Y., He, B. & Chen, Y. 2020 Pressure prediction and abnormal working conditions detection of water supply network based on LSTM. *Water Supply* **20** (3), 963–974. https://doi.org/10.2166/ws.2020.013.

Yin, J., Deng, Z., Ines, A. V. M., Wu, J. & Rasu, E. 2020 Forecast of short-term daily reference evapotranspiration under limited meteorological variables using a hybrid bi-directional long short-term memory model (Bi-LSTM). *Agricultural Water Management* **242**, 106386. https://doi.org/10.1016/j.agwat.2020.106386.

Yu, Z., Moirangthem, D. S. & Lee, M. 2017 Continuous timescale long-short term memory neural network for human intent understanding. *Frontiers in Robotics* **11**, 1–14. https://doi.org/10.3389/fnbot.2017.00042.

First received 28 April 2021; accepted in revised form 5 August 2021. Available online 17 August 2021