# Time Series Classification based Correlational Neural Network with Bidirectional LSTM for Automated Detection of Kidney Disease

Navaneeth Bhaskar, Suchetha M, and Nada Philip

Abstract—In this paper, we aim to explore the feasibility of salivary analysis for Chronic Kidney Disease (CKD) detection and thereby design an automated mechanism to detect CKD through analysis of human saliva samples. We have implemented an improved deep learning model that combines both a one-dimensional Correlational Neural Network (1-D CorrNN) and bidirectional Long Short-Term Memory (LSTM) network for making accurate predictions. The LSTM network is integrated with the neural model to utilize the capabilities of both these networks to analyze the time-series data. The proposed model is trained and tested with a CKD sensing module. The application of deep learning algorithms helps to improve the detection accuracy as they are capable of discovering the best features from the input data. The proposed method achieved an average accuracy rate of 98.08% for the testing dataset. The results show that the proposed detection module and classification algorithm substantially advance the current methodologies, and provides more accurate predictions compared to conventional methods.

Index Terms—Convolutional neural network, correlation, chronic kidney disease, long short-term memory, salivary diagnosis.
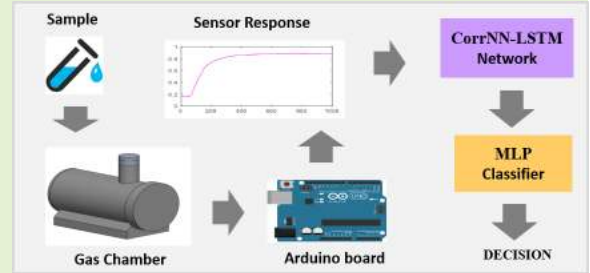
I. INTRODUCTION

ACCORDING to the latest medical report on kidney disease, about 323 million people are affected by CKD globally. Serum urea and creatinine are the widely used biomarkers in CKD diagnosis [1]. Apart from balancing the water level, the kidney's primary functions include the removal of waste products like creatinine and urea from the body. Creatinine is a chemical material released during the metabolic activity of the muscles. Urea is also a chemical substance, which is generated during the processing of proteins. The amount of urea and creatinine in the body increases with an increase in the severity of kidney disease. A blood test is usually performed to diagnose kidney disease, which estimates the levels of either urea or creatinine in the blood serum [2]. Extracting blood samples for diagnosis is not always considered the best approach, as it is an invasive procedure. The biofluid saliva is gaining recognition as a significatory fluid in the diagnostic and prognostic applications [3]. Saliva can be considered as an effective alternative to the blood sample for detecting many diseases as it contains several disease determining biomarkers present in the blood. New

research studies show that the saliva sample can be utilized for CKD diagnosis [1], [4]. Saliva contains urea and creatinine, which are the recognized CKD biomarkers. In the present study, we have examined the levels of urea in the saliva sample to identify kidney disease. There are various methodologies and apparatus available for urea measurement. However, most of the approaches are based on clinical trials, which take more time for analysis because of their intricate detection processes. As the current detection strategies and methods are not ideal for the proposed method, a new detection module is implemented in this work.

In traditional data processing methods, suitable feature extraction algorithms are used prior to the classification process. The need for a separate feature extraction process makes traditional hand-engineered approaches computationally slow and less effective for large data sets and for certain applications. Deep learning networks like CNN can overcome the shortcomings of traditional learning techniques [5]. In the CNN algorithm, the optimal features are extracted automatically. The CNN models are actually designed for image processing, video processing, and other applications where the input is usually a two dimensional (2-D) signal [6]. Recent research works show that CNN can be employed to analyze 1-D signals [7], [8]. However, only a few studies so far have used 1-D CNN algorithm for signal processing applications. Applying CNN for the analysis of 1-D signals was first introduced by Kiranyaz et al. [7]. In their work, the authors have designed a novel 1-D CNN architecture for the characterization of ECG

Navaneeth Bhaskar is with the School of Electronics Engineering, Vellore Institute of Technology, Tamil Nadu, India (e-mail: navaneeth.bhaskar2015@vit.ac.in).

Suchetha M is with the Division of Healthcare Advancement, Innovation and Research, Vellore Institute of Technology, Tamil Nadu, India (e-mail: suchetha.m@vit.ac.in).

Nada Philip is with the Department of Networks and Digital Media, Kingston University, London (e-mail: n.philip@kingston.ac.uk).

signals for detecting heart-related problems. The authors have tested the network with the MIT-BIH arrhythmia database. In a similar study, Acharya et al. [9] adopted CNN for detecting arrhythmia using different intervals of ECG segments. Zhang et al. [10] have designed a 1-D CNN architecture for the analysis of physiological information in humans. In their work, an ensemble CNN model was developed to improve the efficiency and robustness of the individual CNN models. Ince et al. [8] have developed a 1-D version of CNN for identifying the errors in motors. Wu et al. [11] have developed a neural model based on 1-D CNN for analyzing human knee movement. In a recent work, the authors have developed a deep learning model by remodeling the 2-D CNN design to make it feasible for 1-D applications [12]. Their model was used for identifying cardiorespiratory abnormalities, and they attained a prediction accuracy of around 96%.

Although the CNN algorithm functions extremely well, its performance can be further enhanced by making a few changes in its network structure. In this work, we have developed a new Correlational Neural Network (CorrNN) that can provide improved classification performance for automated diagnosis applications. As the name indicates, the proposed network uses a correlation layer instead of the convolution layer for extracting the features. In addition to the correlation network, a bidirectional LSTM network is also used in the model for learning the temporal dynamics in the data. CNN-LSTM combined network was initially introduced for sequence prediction applications with spatial inputs like images or videos [13]. Interestingly, a few recent studies have adopted CNN-LSTM networks for analyzing 1-D signals. Shu et al. [14] have implemented an LSTM based CNN architecture for automated diagnosis of arrhythmia. Similarly, Tan et al. [15] have developed a CNN-LSTM hybrid network for automatically detecting coronary artery disease from ECG recordings. Further, Ay et al. [16] presented a CNN-LSTM network for computerized detection of depression in humans by examining the EEG signals, where they have achieved an accuracy of 99.12%. The authors claim that the proposed model can be applied in hospitals to recognize the depression issues in the patients using the EEG signals. In our architecture, we choose to employ bidirectional LSTMs as they are proven superior in handling temporal sequences. Highly efficient algorithms are needed for processing the medical signals automatically. The proposed correlational network is more efficient compared to conventional CNN and can provide improved classification accuracy.

To the best of our knowledge, this is the first time a correlational neural network is developed with bidirectional LSTM for analyzing the 1-D time-series signals. The rest of this paper is divided into four sections. Section II explains the design structure of the proposed network. Section III describes the material and methods. The performance evaluation and validation results are presented in section IV. The conclusion of the work is given in section V.

II. ARCHITECTURE OF THE PROPOSED MODEL

The proposed deep learning network is designed by modifying the structure of the traditional CNN algorithm. The network architecture of the proposed 1-D CorrNN-LSTM model is shown in Fig. 1. The architecture consists of correlation and sub-sampling layers followed by bidirectional LSTM layers and a fully connected classification layer. The main component in the architecture is the correlational layer. The correlation layer and the sub-sampling layer performs the feature extraction operation to extract the optimal features from the input signal.

A. Correlation Network

We have applied the correlation procedure in place of the convolution operation for extracting the features in the proposed model. Convolution and correlation operations are commonly used in signal processing applications. The convolution process is similar to the correlation process with a small difference. In the convolution operation, the filter is flipped by 180 degrees before performing the convolution process. Nevertheless, the convolution operations referred to in deep learning networks are actually cross-correlation operation because the kernel is not flipped before applying it to the input. Kernels play a crucial role in extracting the features [17]. Flipping of the kernel will invert the kernel values and will lead to a different output. So, flipping of kernels is not performed in the case of neural network learning.

As far as deep learning networks are concerned, the correlation and convolution operations are identical. However, the correlation network presented in this work is a new approach and is different from the existing learning models. In the CorrNN architecture, we have applied the cross-correlation process to extract the optimal features. Correlation is a simple operation where the filter is placed over the input signal, and the sum of the products of the overlapping values are computed. This operation is repeated by shifting the same filter over the entire input signal. The correlation operation between a 1-D input signal I and a kernel k is mathematically represented as:

$$k \circ I(x) = \sum_{i=-n}^{n} k(i)I(x+i) \qquad (1)$$

The correlation process will find the correlation between the two input signals. We get the features from the input signal based on the kernel selected for the correlation process. However, for choosing the kernel size and type, there is no direct method or guideline. The best approach for kernel selection is the trial and error method, where we need to test the network with different kernels and choose the one which gives the most relevant result. Usually, a standard gaussian kernel is used along with 1-D CNN models [18]. As the similarity between the signals is determined in the correlation operation, a kernel procured from the input data can deeply examine the signal's characteristics. Therefore, we have designed and implemented adaptive kernels for the correlation operation. The dimension of the kernel is determined on the basis of the input signal's size for each stage. The details of the adaptive kernel are explained along with the implementation of the proposed network in section III-C.

NAVANEETH *et al.*: TIME SERIES CLASSIFICATION BASED CORRELATIONAL NEURAL NETWORK WITH BIDIRECTIONAL LSTM FOR AUTOMATED DETECTION OF KIDNEY DISEASE
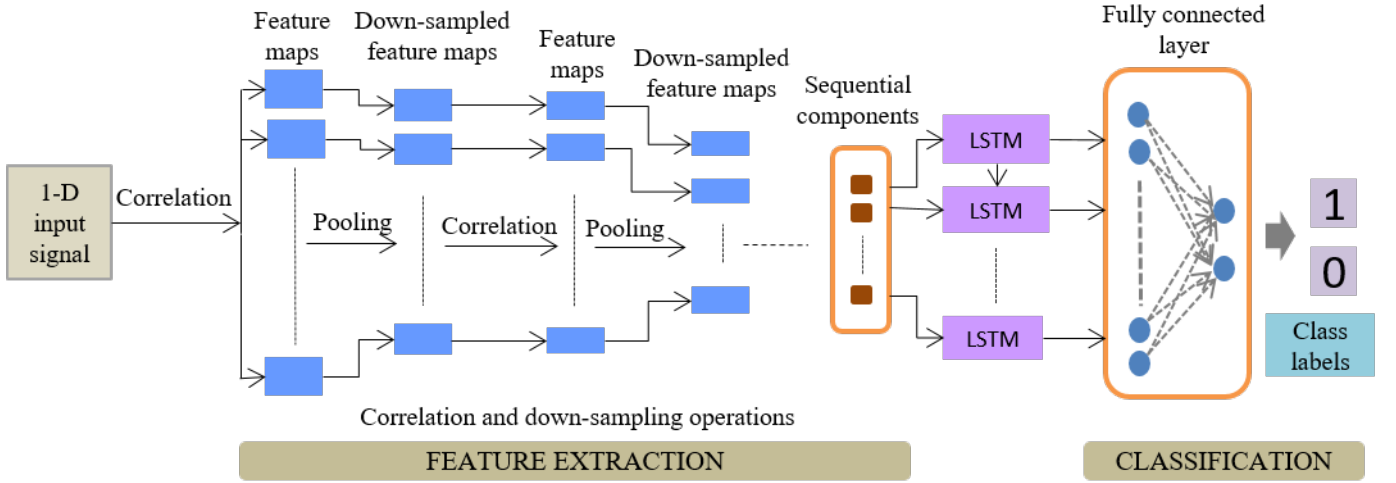
3

Fig. 1: Block representation of the proposed 1-D CorrNN-LSTM network

B. Long Short-Term Memory

LSTM is a deep learning network that is commonly used in the analysis of time-series signals. This network is designed as an improvement to conventional Recurrent Neural Network (RNN). LSTM can deal with the long term dependency problems far better than RNN. LSTM network has memory blocks to handle the issues related to vanishing gradients [19]. This network has the ability to memorize the long-term historical information. The operations on the memory block are managed using adaptive multiplicative gates. The LSTM unit implemented in our architecture comprises of an input gate I_t, an output gate O_t and a forget gate F_t. LSTM can selectively retrieve or neglect any information by using the memory state. Important information is remembered and back-propagated and the unnecessary information is neglected by the network. Bidirectional LSTM is an improvement to conventional LSTM where the forward and backward hidden layers are combined for accessing both the preceding and succeeding data. Bidirectional LSTM can handle the sequential modeling challenge better than conventional LSTM [20].

The primary step in LSTM is to identify the data to be discarded from the cell state. The decision on this is performed by the activation function. A sigmoid activation function is applied in this study. The values of this function vary between -1 and +1. The three gate structures are computed using the following equations:

$$I_t = \sigma(W_i X_t + R_i H_{t-1} + b_i) \quad (2)$$

$$F_t = \sigma(W_f X_t + R_f H_{t-1} + b_f) \quad (3)$$

$$O_t = \sigma(W_o X_t + R_o H_{t-1} + b_o) \quad (4)$$

where b_i, b_f and b_o are the bias gates, W_i, W_f and W_o are the input weights, and R_i, R_f and R_o are the recurrent weights. σ represents the sigmoid activation function, X_t represents the current input and H_{t-1} denotes the previous block output.

The modulated new memory Z_t is computed as:

$$Z_t = tanh(W_t X_t + R_t H_{t-1} + b_t) \quad (5)$$

where W_t and R_t represent the input weight and recurrent weight respectively, and $tanh(\cdot)$ is the hyperbolic tangent function.

The current memory cell M_t is computed as:

$$M_t = I_t \odot Z_t + F_t \odot M_{t-1} \quad (6)$$

where M_{t-1} represents the previous memory cell content and $\odot$ denotes the element-wise multiplication operation.

The output of the LSTM unit is the hidden state H_t and it is estimated as:

$$H_t = O_t \odot tanh(M_t) \quad (7)$$

The LSTM network's output signal is applied to the classification module to perform the final classification task.

C. Fully Connected Classification Layer

The fully connected Multilayer Perceptron layer (MLP) is used after the LSTM network to perform the classification operation to map the extracted features to appropriate classes [21]. Assuming l as the current layer, the input of kth feature signal of this layer is represented as:

$$x_k^l = \sum_{j=1}^{m^{l-1}} corr1D(w_{k,j}^l, z_j^{l-1}) + b_k^l \quad (8)$$

where b_k^l represents the bias, $w_{k,j}^l$ represents the kernel weight from j^{th} signal at (l-1) layer to the kth function at the current layer l, and z_j^{l-1} represents the output of jth function on (l-1). The kth feature signal output is represented as:

$$z_k^l = A(x_k^l) \quad (9)$$

where A represents the activation function.

The weights and bias of the network are randomly assigned during the initial phase. The gradient descent and back-propagation algorithm are introduced to train the model. The initially assigned weights are updated by the succeeding repetitions of these two algorithms. The learning method of the CorrNN includes finding the error E and its gradient $\frac{\partial E}{\partial x}$

Fig. 2: Snapshot of the test setup (Signal acquisition module is shown in the inset)

for evaluating the difference in weights. For reducing the error function, the derivatives of the error are estimated with reference to its weights.

$$\frac{\partial E}{\partial w_{k,j}^l} = \Delta w_{k,j}^l \qquad (10)$$

$$\frac{\partial E}{\partial w_{k,j}^l} = \frac{\partial E}{\partial x_k^l} \frac{\partial x_k^l}{\partial w_{k,j}^l} \qquad (11)$$

By taking the derivative of the activation function, $\frac{\partial E}{\partial x_k^l}$ is obtained as:

$$\frac{\partial E}{\partial x_k^l} = \frac{\partial E}{\partial z_k^l} A'(x_k^l) \qquad (12)$$

Finally, the weights are updated as follows:

$$w_{k,j\,new}^l = w_{k,j}^l + \gamma \Delta w_{k,j}^l \qquad (13)$$

where γ is the learning rate.

The forward and backward propagation procedures are repeated till the error is minimized. The error value decreases in each iteration step. After the training process, the testing dataset is given as input to the trained model.

III. MATERIAL AND METHODS

A. Signal Acquisition Module

We have implemented a new method for urea detection. The concentration of urea in the saliva sample is determined by converting it into ammonia gas. For converting urea to ammonia, we have adopted the conventional enzymatic transformation process [22]. Urease enzyme supplied by Calzyme Laboratories is used for this conversion reaction. This enzyme has the ability to hydrolyze urea molecules into ammonia. The conversion reaction is to be carried out in a closed chamber. So, we have specially designed a gas chamber with an internal capacity of 212ml. The gas chamber is designed with an input opening at the top. The enzyme needed for the conversion is placed under the input valve in a small petri dish. A semiconductor-based MQ-137 sensor, which is very responsive to ammonia gas is used to measure the gas generated inside the chamber [23]. MQ-137 ammonia sensor developed by Hanwei

Electronics is used in the sensing model. The reactive material of this sensor is a stannic oxide layer on its surface. This sensor requires a circuit voltage of 5V for its operation. A heater voltage of 5V is also given to the sensor for providing the required temperature. Figure 2 shows the photograph of the testing module and the experimentation. To analyze the static and dynamic properties of the sensor, it is initially tested with different concentrations of ammonia gas. Ammonia gas is introduced into the gas chamber using a micro-syringe and the behavior of the gas sensor is monitored. The sensor's response depends on the ammonia gas concentration inside the conversion chamber. It is seen that the output voltage of the gas sensor changes corresponding to the ammonia gas variations inside the chamber. The sensor showed linear behavior with increasing concentrations of ammonia gas. We have carried out our preliminary analysis in a controlled environment by operating the sensor in its conventional working state to reduce the impact of humidity and all interfacing gases.

B. Saliva Collection and Testing

The samples are collected from 104 participants for the experimental analysis. Subjects selected for the study includes 64 CKD patients and 40 healthy individuals. Saliva samples are collected in a 5ml graduated cylinder using the spitting approach. 1ml of saliva is collected from each subject with the assistance of medical professionals. The sample examination is done by dropping 1ml of the sample through the inlet valve using a dropper. Hydrolysis of urea occurs within the chamber when the sample is dropped onto the urease enzyme. Ammonia gas is generated in this reaction process. The sensor's conductivity varies according to the ammonia gas produced inside the chamber. This difference in the conductivity is transformed into a voltage with the help of a simple electrical circuit. An Arduino Uno microcontroller board is connected to the sensing module for acquiring the sensor reading and for converting the signal. The sensor signal is recorded for every 0.1 seconds, and it is recorded for 100 seconds so that the output is obtained for a longer period. The sensor response waveform obtained for a healthy and CKD test sample is illustrated in Fig. 3. In this work, we are not manually extracting the features for the classification process. The features are automatically extracted by the proposed neural network model. Therefore, even the minute variations in the sensor response will be considered for computing the feature maps. The proposed research work is conducted in compliance with the ethical standards of the Declaration of Helsinki. The objective of the study was explained to the participants, and their consent was taken before the experimentation.

C. Feature Extraction and Classification

The raw signal obtained from the sensor is first normalized before giving it to the learning networks. Normalization of the sensor response is done to ensure that the amplitude of the sensor is within a specified level. The following equation is used to normalize the signal:

$$s(n) = \frac{x(n) - \frac{1}{P}\sum_{n=1}^{P} x(n)}{\sqrt{\frac{1}{P}\sum_{n=1}^{P}(x(n) - \frac{1}{P}\sum_{n=1}^{P} x(n))^2}} \qquad (14)$$
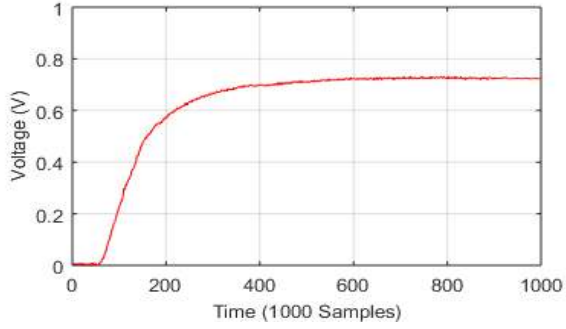
(a) Sensor response graph for a healthy test sample
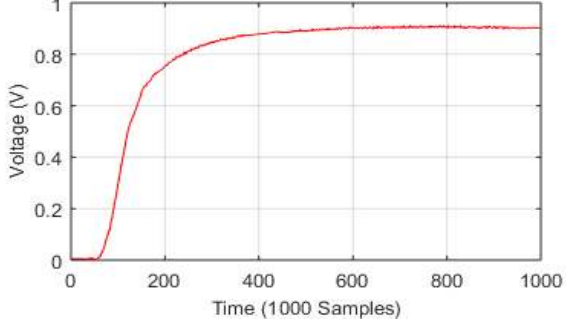
(b) Sensor response graph for a CKD patient

Fig. 3: Graphs displaying the sensor output signal for a healthy and CKD test sample

TABLE I: DETAILS OF NETWORK STRUCTURE AND SIGNAL DIMENSION

Layers	Input signal	Kernel size	Correlated signal	Output signal
1	1×1000	1×1000	1×1999	1×399
2	1×399	1×399	1×797	1×159
3	1×159	1×159	1×317	1×63
4	1×63	1×63	1×125	1×25
5	1×25	1×25	1×49	1×9
6	1×9	1×9	1×17	1×3

Fig. 4: Bidirectional LSTM operation for three time steps

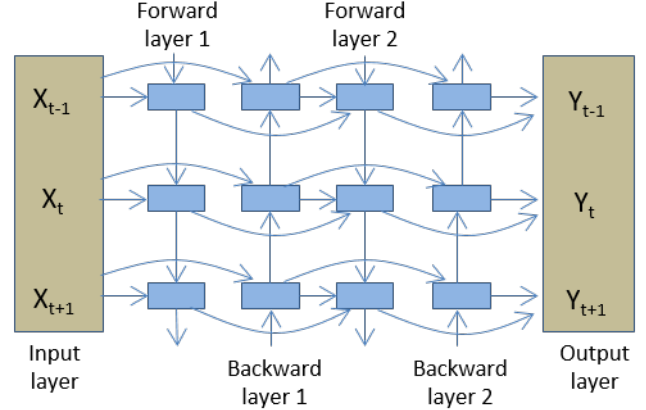where x(n) is the signal sampled to 1000 points and P is the number of data points in the input signal.

The normalized signal is correlated with the kernel filter to obtain the most desirable features from the sensor response. The proposed network has six correlation and pooling stages. Initially, we have carried out the correlation process with a predefined gaussian kernel. Subsequently, we tested the network with our time-series adaptive kernels. It is observed that the correlation works better with our adaptive kernels as the correlation operation measures the similarity between the kernel and the input signal.

As the size of the input signal is 1×1000, the kernel for the first stage should have a dimension of 1×1000. To achieve this, we have taken the average of sensor readings recorded for all the 40 healthy cases, and this averaged signal is used as the kernel. For deriving the kernel for the second correlation stage, the kernel used for the previous stage is correlated with itself. The correlation of kernel is performed to make the kernel and correlated signal characteristics the same. The correlation will produce a signal with a size of 1×1999. The correlated kernel signal is divided into multiple sections by applying the sliding window algorithm with segment size five. After the segmentation process, 399 subsections will be generated. The peak value from every section is selected for deriving the kernel function. Therefore, the dimension of the kernel for the second stage will be 1×399. Next, for deriving the kernel signal for the third stage, the kernel applied for the second stage is correlated with itself. The correlated signal is then split into different segments by retaining the segment size as

five, and the maximum value from each section is selected. Hence, the size of the kernel obtained for the third stage will be 1×159. The same steps are repeated for deriving the kernel signals for the remaining stages. Consequently, kernel signals with proportions of 1×63, 1×25 and 1×9 are obtained for the last three correlation layers, respectively.

The kernel signal is slid over the input signal by keeping the stride value as one. Simultaneously, the sum of the products of the overlapping values is computed. The first correlation process will generate feature maps with 1999 values. The segment size is set as five for all the pooling procedures. Thus, after the first pooling operation, the size of the feature map will be 1×399. The output of the first correlation layer will serve as the input for the second correlation layer. This signal is correlated with the second kernel to produce layer 2. This is followed by the second pooling process, which decreases the feature map dimension to 1×159. As our architecture is designed for six stages, the same procedure is repeated until layer 6. The kernel size and the signal dimensions are presented in Table I. A reduced feature map with a size of 1×3 is obtained after the final correlation operation.

Optimal features are obtained after performing the correlation and pooling operations. Extracted features are given to the bidirectional LSTM layer. The memory blocks in the LSTM network is used to learn the previous input for a long duration [16]. The data flow is controlled using the three gates. The input gate decides what information should be saved in the memory block. The output gate makes decisions on the data stored in the memory block. The amount of information to be saved or discarded in the memory block is decided by the forget gate. The data which is temporarily saved are fused
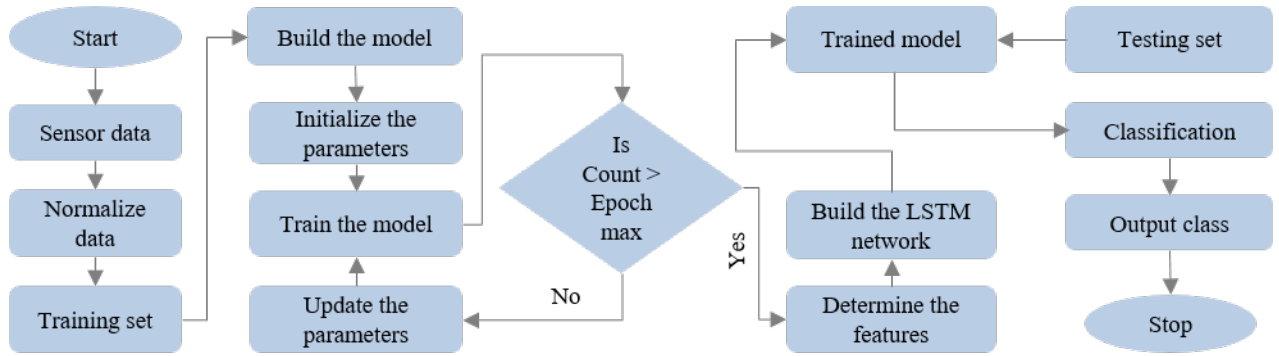
Fig. 5: Flowchart of the proposed model

with the previous memory state. The bidirectional LSTM will process the data in both directions by using individual layers. The input data and the values at every succeeding layer are fed to both forward and backward layer. The bidirectional LSTM operation for three time steps ($t-1$, t and $t+1$) is shown in Fig. 4. There are no hidden-to-hidden connections between the layers. The output layer of the network receives the data by associating the forward and backward layers. The output of the LSTM module is given to the final layer of the model, which is the MLP layer. The MLP performs the classification task and makes the prediction. Training of the network is done using back-propagation and the resultant error is reduced using the gradient descent method.

IV. RESULTS AND DISCUSSION

The normalized sensor output signal is given to the proposed network for the analysis and classification. The last pooling layer feeds into the bidirectional LSTM network. The output of the LSTM layer is fed to the classification model. The flow diagram of the steps involved in the proposed learning model is illustrated in Fig. 5. In this work, the classification problem is to identify CKD and healthy samples. This is a binary classification as there are only two classes i.e CKD class or healthy class. The classifier needs the training to identify how the given samples are associated with the particular class. The algorithm can identify whether CKD is there or not for a particular sample once it is trained. The proposed network is built and trained in Matlab 2017b environment.

The performance of the designed network is examined by evaluating the performance parameters. For comparing the performance of the proposed model, we have implemented different traditional algorithms. The network algorithms analyzed in this work are SVM classifier with Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) algorithms, RNN, Shallow CNN, 1-D CNN-MLP algorithm with six layers and CNN-SVM hybrid network with different kernels. All these algorithms are tested with the same input samples. SVD and PCA are the traditional feature extraction techniques. The features extracted by these algorithms are classified using the SVM classifier. RNN is commonly used in time series prediction. RNN network transforms the independent activations into dependent ones by offering the same weights and biases to every layer of the network. CNN

with a single hidden layer is the shallow CNN model. In the CNN-MLP algorithm, CNN works on feature extraction and MLP does the classification task. The CNN-SVM algorithm is implemented with linear kernel, polynomial kernel, sigmoid kernel and gaussian kernel.

To evaluate the efficiency of the learning models, the accuracy rate is calculated based on the following formula:

$$Accuracy = \frac{N_P}{N_T} \times 100\% \qquad (15)$$

where N_P and N_T represent the correctly predicted samples and total samples, respectively.

Other parameters considered for performance evaluation are precision, sensitivity, specificity, False Negative Rate (FNR), False Discovery Rate (FDR), Matthews Correlation Coefficient (MCC), Negative Predictive Value (NPV), F1 score, False Positive Rate (FPR) and Misclassification Rate (MR). Ten-fold cross-validation is performed to estimate the model performance. Accordingly, the samples were divided into ten equal parts. Out of the ten subsamples, nine are used for training the model, and the remaining one subsample is applied as a test sample for validating the performance. This process is repeated ten times by changing the test dataset. The performance parameter values are evaluated after every iteration. Finally, the average of these ten iterations is estimated to assess the performance of the model. MSE values of 0.145 and 0.148 are obtained for the proposed 1-D CorrNN-LSTM algorithm and traditional 1-D CNN algorithm respectively. Figure 6 shows the comparison of the MSE plot.

The measured parameter values are summarized in Table II. The features extracted through SVD and PCA are separately classified using the SVM classifier. The PCA-SVM and SVD-SVM models have classified the data with an average accuracy rate of 88.46% and 89.42%, respectively. The RNN model classified the samples with 92.57% accuracy. As the features are automatically extracted in CNN-based algorithms, they have performed better than SVD-SVM and PCA-SVM algorithms. The conventional CNN-MLP algorithm attained a classification accuracy of 96.49%. The performance of CNN got better when an SVM classifier is used in place of MLP. The CNN-SVM with gaussian kernel has performed slightly better than other kernels. The proposed correlation network achieved the highest accuracy rate compared to all other networks analyzed in this study. The confusion matrix obtained for the
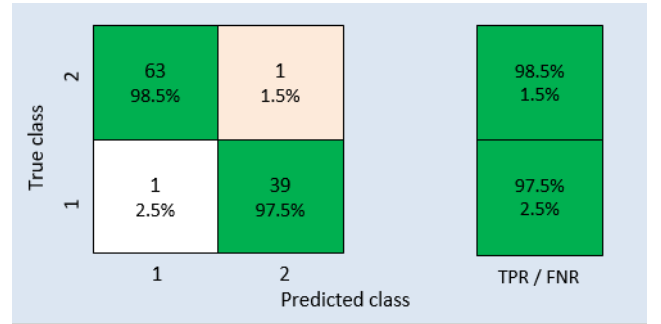
TABLE II: PERFORMANCE MEASURES OF DIFFERENT NETWORK ALGORITHMS

Algorithms	Accuracy	Sensitivity (%)	Specificity (%)	Precision (%)	FPR	FNR	NPV	FDR	F1 Score	MCC	MR
SVD-SVM	89.42	92.19	85.01	0.908	0.15	0.078	0.872	0.092	0.915	0.776	0.106
PCA-SVM	88.46	92.19	82.5	0.894	0.175	0.078	0.868	0.106	0.908	0.755	0.115
RNN	92.57	94.31	96.18	0.958	0.038	0.057	0.947	0.041	0.951	0.905	0.074
Shallow CNN	95.05	94.3	95.74	0.953	0.042	0.057	0.947	0.046	0.948	0.91	0.049
CNN-MLP	96.49	98.04	95.24	0.943	0.048	0.019	0.984	0.057	0.962	0.929	0.035
CNN-SVM (Linear)	97.27	96.08	98.41	0.98	0.015	0.039	0.968	0.02	0.97	0.945	0.027
CNN-SVM (Sigmoid)	96.42	96.08	96.83	0.961	0.032	0.04	0.968	0.039	0.96	0.945	0.036
CNN-SVM (Gaussian)	97.37	98.03	96.82	0.961	0.031	0.019	0.987	0.038	0.977	0.947	0.026
CNN-SVM (Polynomial)	97.14	98.43	95.12	0.969	0.048	0.015	0.975	0.03	0.977	0.939	0.029
Proposed 1-D CorrNN-LSTM	98.08	98.44	97.5	0.984	0.025	0.016	0.975	0.016	0.984	0.959	0.019
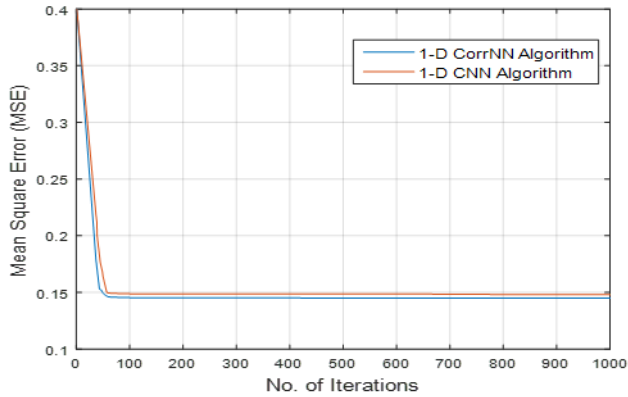
Fig. 6: Comparison of mean square error plot of CNN algorithm and proposed algorithm

Fig. 7: Confusion matrix obtained for the proposed model

proposed classification model is shown in Fig. 7. The 1-D CorrNN-LSTM network has effectively classified the samples as a 'healthy' or 'diseased' sample with an average accuracy of 98.08%. The misclassification rate of the proposed algorithm is 0.019. The use of bidirectional LSTM in the network has improved the functionality of the model to classify the signals with varying sequence lengths.

We have evaluated the computing time of the proposed algorithm and it is compared with the conventional CNN model. As the computing time of the algorithm depends on the processor speed and configuration of the system, it is important to consider the specifications of the system used for the analysis. In our experimentation, we have done the analysis on an Intel i7-7700HQ four-core 2.8GHz processor laptop computer with 16GB DDR4 RAM. As we are using a larger kernel for the correlation process, the feature extraction time will be more for the CorrNN model compared to CNN. The time taken for feature extraction is 1.146s for the proposed network, which is a little high value compared to that of the CNN model. As the difference in the computing speed is very less, it won't affect the classification performance much. The total time taken for computation by the proposed algorithm is 2.427s. The conventional CNN-MLP has taken only 2.174s

for computation.

To verify the test results, clinical validation is also carried out. Validating the results helps us to ascertain how good the proposed networks can classify a person being examined is a kidney patient or not. In clinical practice, kidney patients are detected by determining the Glomerular Filtration Rate (GFR) [24]. The GFR of all the subjects was measured using the clinical analysis. This test is the widely recognized method to diagnose CKD. A non-CKD person will have a GFR higher than 90mL/min. Out of the 104 samples tested, the proposed method has classified 64 samples under the CKD class and 40 samples under the healthy category. This model misclassified only two samples. The GFR of 63 subjects who have been classified under the CKD category by the proposed model, had GFR less than 90mL/min. 39 healthy samples are correctly classified under the healthy category by the proposed model. Further, we have performed a t-test and obtained a p-value of less than 0.05. This confirms that the results are statistically significant. The experimental outcome shows that the proposed network has the potential to be used for automated detection applications. Automated disease detection models like this are very useful in a country like India, where doctors and paramedical staff are facing immense pressure in managing a disproportionately larger number of patients.

V. CONCLUSION

In this work, we have developed a novel sensing model and a hybrid deep learning network that combines both Cor-

rNN and bidirectional LSTM for detecting kidney disease. The CorrNN will learn the local properties and the LSTM network learns the sequences from the signals. To explore the capabilities of the proposed technique, experiments are carried out with the hardware sensing module. The saliva-based approach presented in this work will be more acceptable to patients as the sample extraction method is stress-free and painless. The performance of the implemented networks is evaluated by determining the various performance parameters. The proposed hybrid model attained promising performance in the automated detection of CKD. The prediction accuracy rate achieved by the proposed method is 98.08%, and it is higher than other traditional methods. This study shows that the integration of LSTM with the neural network model can provide better efficacy when processing 1-D signals. Further, we have performed the clinical validation and the result shows a good correlation with our experimental results. Future research will focus on the VLSI circuit implementation of the proposed model.

REFERENCES

[1] S. Chojnowska, T. Baran, I. Wilinska, P. Sienicka, I. Cabaj-Wiater, and M. Knas, "Human saliva as a diagnostic material," *Advances in medical sciences*, vol. 63, no. 1, pp. 185–191, 2018.
[2] P. Celec, L. Tothova, K. Sebekova, L. Podracka, and P. Boor, "Salivary markers of kidney function-potentials and limitations," *Clinica Chimica Acta*, vol. 453, pp. 28–37, 2016.
[3] T. A. V. Pham, "Validation of the salivary urea and creatinine tests as screening methods of chronic kidney disease in vietnamese patients," *Acta Odontologica Scandinavica*, vol. 75, no. 8, pp. 551–556, 2017.
[4] B. S. Bagalad, K. Mohankumar, G. Madhushankari, M. Donoghue, and P. H. Kuberappa, "Diagnostic accuracy of salivary creatinine, urea, and potassium levels to assess dialysis need in renal failure patients," *Dental research journal*, vol. 14, no. 1, p. 13, 2017.
[5] P. Zhang, X. Wang, W. Zhang, and J. Chen, "Learning spatial–spectral–temporal EEG features with recurrent 3D convolutional neural networks for cross-task mental workload assessment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 1, pp. 31–42, 2019.
[6] R. Zhang and S. Cao, "Real-time human motion behavior detection via CNN using mmwave radar," *IEEE Sensors Letters*, vol. 3, no. 2, pp. 1–4, 2018.
[7] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2016.
[8] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks." *IEEE Trans. Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.
[9] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam, "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network," *Information sciences*, vol. 405, pp. 81–90, 2017.
[10] J. Zhang, S. Li, and R. Wang, "Pattern recognition of momentary mental workload based on multi-channel electrophysiological data and ensemble convolutional neural networks," *Frontiers in neuroscience*, vol. 11, p. 310, 2017.
[11] H. Wu, Q. Huang, D. Wang, and L. Gao, "A CNN-SVM combined model for pattern recognition of knee motion using mechanomyography signals," *Journal of Electromyography and Kinesiology*, vol. 42, pp. 136–142, 2018.
[12] D. Bhagya and S. Manikandan, "Speed of sound-based capnographic sensor with second-generation CNN for automated classification of cardiorespiratory abnormalities," *IEEE Sensors Journal*, vol. 19, no. 19, pp. 8887–8894, 2019.
[13] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified-lstm model for continuous sign language recognition using leap motion," *IEEE Sensors Journal*, 2019.
[14] S. L. Oh, E. Y. Ng, R. San Tan, and U. R. Acharya, "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats," *Computers in biology and medicine*, vol. 102, pp. 278–287, 2018.
[15] J. H. Tan, Y. Hagiwara, W. Pang, I. Lim, S. L. Oh, M. Adam, R. San Tan, M. Chen, and U. R. Acharya, "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals," *Computers in biology and medicine*, vol. 94, pp. 19–26, 2018.
[16] B. Ay, O. Yildirim, M. Talo, U. B. Baloglu, G. Aydin, S. D. Puthankattil, and U. R. Acharya, "Automated depression detection using deep representation and sequence learning with EEG signals," *Journal of medical systems*, vol. 43, no. 7, p. 205, 2019.
[17] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors Journal*, 2019.
[18] B. Navaneeth and M. Suchetha, "A dynamic pooling based convolutional neural network approach to detect chronic kidney disease," *Biomedical Signal Processing and Control*, vol. 62, p. 102068, 2020.
[19] P. Kaushik, A. Gupta, P. P. Roy, and D. P. Dogra, "Eeg-based age and gender prediction using deep BLSTM-LSTM network model," *IEEE Sensors Journal*, vol. 19, no. 7, pp. 2634–2641, 2018.
[20] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2278–2288, 2018.
[21] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2015.
[22] N. Bhaskar and M. Suchetha, "An approach for analysis and prediction of CKD using deep learning architecture," in *2019 International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2019, pp. 1660–1664.
[23] T. Saidi, O. Zaim, M. Moufid, N. El Bari, R. Ionescu, and B. Bouchikhi, "Exhaled breath analysis using electronic nose and gas chromatography–mass spectrometry for non-invasive diagnosis of chronic kidney disease, diabetes mellitus and healthy subjects," *Sensors and Actuators B: Chemical*, vol. 257, pp. 178–188, 2018.
[24] X. Zheng, W. Wei, Q. Huang, S. Song, and G. Huang, "Automated region of interest detection method in scintigraphic glomerular filtration rate estimation," *IEEE journal of biomedical and health informatics*, vol. 23, no. 2, pp. 787–794, 2018.

Navaneeth Bhaskar received the BE and M.Tech degree from Visvesvaraya Technological University, India. He is currently pursuing the Ph.D degree in signal processing at Vellore Institute of Technology, Chennai Campus, India. His research interests includes biomedical signal processing, sensor designs, IOT and machine learning techniques.

Suchetha M received the Ph.D degree in biomedical signal processing from Anna University, Chennai, in 2014. She is currently working as Associate Professor at the Division of Healthcare Advancement, Innovation and Research, Vellore Institute of Technology, India. Her areas of interest includes biomedical signal processing , EMD and machine learning.

Nada Philip received the Ph.D degree in mHealth from the Faculty of Science, Engineering and Computing at Kingston University, in 2008. She is currently working as Associate Professor in the field of Mobile health at Kingston University, London. Her research interests include multimedia communication, networking and information technology in medical applications.