Deep learning approach on tabular data to predict early-onset neonatal sepsis

Redwan Hasif Alvi, Md. Habibur Rahman, Adib Al Shaeed Khan and Rashedur M. Rahman
Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh

ABSTRACT
Neonatal sepsis that is a major threat for maternal and neonatal health worldwide. In this work we design non-invasive, deep learning classification models for predicting accurately and efficiently the early-onset sepsis in neonates in Neonatal Intensive Care Units. By non-invasive, it means that no external instrument or foreign body is introduced when taking data for the classifier. Moreover, the data collected for the purpose of predicting and classifying subjects with neonatal sepsis is in the form of tabular, structured data. The deep learning classification models we design and propose in this are known for working with time series, sequential or image data. Hence, the objective of the current research is to propose such a model that makes use of the powerful tools inherent in Neural Networks for pattern recognition, and use them to outperform traditional machine learning algorithms to detect early-onset neonatal sepsis. Real life neonatal sepsis data samples from two different hospitals are used (Crecer’s Hospital Centre in Cartagena-Colombia and Children’s Hospital of Philadelphia) to make the evaluation of the Neural Networks as authentic as possible.

ARTICLE HISTORY
Received 21 June 2020
Accepted 25 October 2020

KEYWORDS
Machine learning; convolution neural networks; LSTM; tabular data; neonatal sepsis

Introduction
Neonatal sepsis is a form of blood infection that affects neonates under 28 days of age. Neonatal sepsis is classified into two classes: early-onset sepsis (EOS) and late-onset sepsis (LOS). Early-onset sepsis corresponds to infants at or before 72 h or birth, while late-onset sepsis corresponds to infants affected by neonatal sepsis after the 72 h mark (Singh et al., 2020). Specifically, it is related to the occurrence of bacterial bloodstream infection (BSI) in a newborn baby (such as pyelonephritis, meningitis, gastroenteritis, or pneumonia) in fever environment.

Neonatal sepsis is considered to be a global threat for maternal and neonatal health. A good number of deaths is reported in the children under five years of age, and those deaths take place during the neonatal period, and unfortunately neonatal sepsis is the cause of 26% of those deaths (Adatara et al., 2019).
Infection found to be affirmative of bacteria and fungi in the blood or cerebrospinal fluid (CSF) culture, and viruses that manifest within the initial 3 days of birth in pre-term is recognized as early-onset neonatal sepsis (Gómez et al., 2018). The blood or CSF culture checked for and bacterial or fungal infection. This method is a limiting factor in detecting sepsis as the rate of detection depends on the rate of clinical examination and length of the test. Early detection contributes to fast diagnosis, which has been shown to minimize treatment delay, and reduce death rate in affected patients (Nguyen et al., 2007).

By using readily accessible electronic medical records (EMR), we can achieve substantial data that we can evaluate and use for developing non-invasive classification models for prediction. These models can be used for early detection of sepsis cases in infants. Analysis of such broad data can reveal trends and fatal symptoms in infants that can help predict and avoid early-onset neonatal sepsis. It is difficult to clinically diagnose sepsis in infants. Antibiotic therapy is usually initiated when doctors suspect sepsis and sometimes before the laboratory test results are in hand. Machine learning classifier could be an outstanding assistant for the service provides in this domain.

Globally, it is a major concern for both maternal and neonatal health. Sepsis is a major contributing factor to the mortality rate in infants below the age of 5, out of which a sizeable portion of death takes place during the neonatal period (icddr,b - Maternal and neonatal health, 2020).

On the other hand, neural networks are traditionally used complex scenarios involving computer vision, image classification, time series data and sequential data. Neural Networks (especially convolutional and recurrent neural networks) are not used for static tabular data, which is more traditionally associated with Logistic Regression, K-Nearest Neighbour, Support Vector Machines (SVM) and Tree classifiers such as Random Forest. We could not find any mainstream work or research based on the use of convolutional nets and recurrent nets used to classify tabular data. Hence, it added additional motivation to build a convolution neural network and recurrent neural network to train and test on static data and see if these powerful classifiers can best the traditional ones.

The challenge in most machine learning and deep learning scenarios is getting appropriate data for the model to use. Previous work done on neonatal sepsis with machine learning mostly use tabular data. Hence, using non-time series data for classifier models that are used for sequential data types is extremely rare. So, we have to manipulate the data as to be able to make the models accept it for the purpose of training, testing and predicting.

The research contributions are as follows:

(1) Two models are developed to implement a non-invasive approach to detect sepsis in neonates.

(2) The sepsis dataset is in tabular format. The array of data is transformed into images that are suitable to feed into the convolution neural network. Each row in the tabular data represents an individual sample. Each sample is converted into a three-dimensional matrix of pixels, which ultimately represents itself as an image file and could be trained by convolutional and recurrent neural networks. Model 1 and 2 are developed based on those networks. The novelty in this is that static, non-time series data is fed into convolution and recurrent neural networks; these
models are known to excel with time-series data and are not typically used for static tabular data.

(3) Using different hyper-parameters, model 1 and 2 achieved promising results. No feature selection is carried out for model 1 and 2 as these were able to utilize the entire dataset. Feature selection is used for model 1. Finally, three models are applied on Lopez-Martinez data set and another recent neonatal sepsis data set of Children’s Hospital of Philadelphia, USA reported by Masino et al. (2019). Experimental results proved the efficacy of the models compared to the traditional models.

**Related work**

In recent years a good number of research papers reported the use of machine learning algorithms on neonatal sepsis. Mani et al. (2014) implemented many traditional predicting models to test 299 infants for late-onset neonatal sepsis. Highly predictive features are selected using feature selection algorithms. The study achieved an AUC of 78%. Some of the models used include Lazy Bayesian Rules, regression trees, and support vector machines Griffin et al. (2005) used multivariable logistic regression to detect sepsis in newborns, and reported an AUC of 82%. Calvert et al. (2016) developed a high-performance model to detect early-onset sepsis. The model took nine vital symptomatic signs which include white blood cell count, pH, blood oxygen saturation, systolic blood pressure, pulse pressure, heart rate, temperature, respiration rate, and age of subjects as predictive features. A mean AUC of 92% was achieved in this study. Desautels et al. (2016) used a combination of patient data such as vitals, age and peripheral capillary oxygen saturation, and applied them to the classification model developed by Calvert (Calvert et al., 2016) to report a mean AUC of 88% Horng et al. (2017) presents a model that identifies patients with sepsis in the emergency room. The study achieved an AUC of 85%. Kam and Kim (2017) implemented deep learning methodologies to develop a detection classifier and compared the model to regression techniques. The classifier scored 92.9% for AUC. López-Martínez et al. (2019) presents an artificial neural network classifier to predict early-onset neonatal sepsis. The dataset used in this study contains 555 samples, and it has an imbalanced class distribution. The model achieved an AUC 92.5%. We used this work as the baseline for our initial study since we used the same dataset as our main source of data. Masino et al. (2019) developed and evaluated machine learning models to detect sepsis in newborns hospitalized at least four hours before they are suspected clinically. We also use a second dataset from this study in order to have a versatile selection of data for our classifiers.

We used the dataset published in the Lopez-Martinez study to improve upon their results and publish it in our previous work (Alvi et al., 2020). In that study, we got an AUC of 99.8% and an accuracy of 98.2% using a feed-forward neural network with 3 hidden dense layers. We implemented feature selection based on strength score of each independent variable in relation to the class variable in order to select the variables that allowed for the vast improvement in result using the same dataset.

Lastly, we developed two new models to implement a non-invasive approach in detecting early-onset neonatal sepsis. This study is an extension to the work we published previously (Alvi et al., 2020). In the current work, our goal is to do further research on
early-onset neonatal sepsis using deep learning models. Two new models we build in this study are convolutional and recurrent neural networks. Unlike the previous study, we do not use any feature selection or data balancing techniques in this work. Instead, we build deep learning neural networks that take in static, tabular data as input and treat them as sequential, time series data. In addition to the dataset publish by Lopez-Martinez, we also use the dataset used by Masino in this work. This allows us to observe how our models perform on multiple datasets; something we could not check in our previous work. Our models achieve the best performance out of all the previously mentioned studies. Similar to Masino, our model can be used in expert systems neonatal intensive care units to predict early-onset sepsis as soon as the data is provided. This can be 4 h or more prior to clinical suspicion.

Development

For our study, we build two deep learning neural network classification models to predict cases of early-onset sepsis in neonates. This is done at an earlier stage than clinical diagnosis could. Additionally, a number of other machine learning classification models are also implemented to compare the results of our Neural Net models. We use the Sklearn library (Pedregosa et al., 2012), Keras framework and Azure Machine Learning studio to develop these models. Moreover, data visualization and normalization, data pre-processing and any data balancing techniques are implemented using Python programming language.

Design

We build two neural networks (CNN, LSTM-RNN) as the main models to classify subjects based on whether or not they are affected by early-onset neonatal sepsis. For comparison, we develop other machine learning classifiers. In this section, we will discuss the methodologies for building and training each model for our work. Models that do not use neural network architectures (k-nearest neighbour, Tree-based classifiers, support vector machines) will not be explored in detail because these models did not require special attention during development and implementation phase.

To have a fair distribution and training and testing data, we performed a Train/Test split on the Lopez-Martinez dataset with a 07:03 ratio. Thus, we used 70% of the dataset to train the models, and 30% (167 samples) of the dataset for validation, and to test the models for accuracy.

Convolution neural network

Convolution Neural Networks (CNN) are a special form of multi-layer feed forward neural network that are widely used in tasks related to image classification, segmentation, object detection and computer vision. In addition to having fully-connected layers similar to an artificial neural network, CNNs also contain two additional types of layers: convolutional and pooling layers.

The convolutional layers are used to analyse an input and extract a feature map by using a set of filters. This feature map is then provided to the next layer, a max pooling
layer, to decrease the size of the feature map that is generated from the earlier layer. Decreasing the size of the feature map is useful because it reduces the dimensionality of the feature map to its most essential information, and helps reduce the processing time. This leads to faster convergence rate which improves generalization performance (Nagi et al., 2011).

For our research, we have developed a standard convolution neural network with one convolution layer and one max pooling layer followed by a fully-connected dense layer. Figure 1 shows a general visualization of our CNN architecture. Although the input data is same as all the other models, we feed the input differently to the convolution neural network. More details can be found in the Data Representation portion of this section.

After the data goes through the convolution and max pooling processes, the condensed feature map outputs pass on to the series of fully-connected layers. By these layers our model flattens the maps together in order to compare the probabilities of the features occurring together. This continues until the best performance is achieved.

Flattening the feature maps is an important step because our convolutional model has the fully-connected layers. These layers are basically representatives of an artificial neural network. Since ANN models only take one-dimensional arrays as input, we have to flatten our three-dimensional matrix of pixels in a one-dimensional array. Thus, ‘flatten’ creates our input for the fully-connected dense layer of the model.

Our CNN model architecture can be represented as

\[
\text{Input Layer} \rightarrow \text{Convolutional Layer} \rightarrow \text{Max Pooling Layer} \rightarrow \text{Dense Layer} \rightarrow \text{Output Layer}
\]

where the shape of the input layer is the shape of the three-dimensional matrix. The convolutional layer consists of 32 filters; each filter size is $3 \times 3$. We take activation function as ReLU for all layers (except for the output layer) because of its popularity as well as computational speed and efficiency. Moreover, using ReLU allows us to not have to use a separate layer for normalization.

We use a max pooling layer of size $2 \times 2$ and flatten the output feature map into an artificial neural network layer of 100 nodes. Our output layer contains 2 nodes for classification with ‘softmax’ activation function in order to give a probability for the possible classification. Dropout is used to prevent over-fitting in CNN as otherwise the model can easily overfit relatively small training data (Wu & Gu, 2015). We apply a dropout of

![Figure 1](image-url)
0.5, which means that it drops out 50% of the weights during the learning process. Categorical cross-entropy is used as the loss function similar to the artificial neural network.

For the convolutional neural network model, we try three optimizing functions: (i) stochastic gradient descent (sgd), (ii) Adam optimizer and (iii) Adadelta optimizer. Optimizers (i) and (ii) are same as used in the ANN model in our previous study (Alvi et al., 2020). However, for our convolution model, we find that sgd gave the worst result while Adadelta yielded the best results. Adadelta optimization is a stochastic gradient descent method based on the adaptive learning rate per dimension. It solves the setback of continual decay of learning rates throughout training as well as removes the need for manually selecting a global learning rate (Adadelta, 2020). Table 1 shows the hyper-parameters used for the CNN.

Proper loss functions and activation functions highly determine the accuracy of the model. Hence, we tested multiple such functions to select the one best suited for our model. Figure 2 shows the visualization of the pooling filters used in the convolution neural network.

**Data representation**

The convolution neural network model accepts input as images. However, the data we have is in the form of an array. Since computers read images represented as three-dimensional matrix of pixels, we reshape our input array as a three-dimensional matrix (length, width, colour channel).

Thus, we transform our array of data into images that are suitable to feed into our convolution neural network. Each row in the tabular data represents an individual sample. So, we convert each sample into a three-dimensional matrix of pixels, which ultimately represents itself as an image file. We also separate the X (input) and Y (output) from the data, and convert Y into categorical form, transforming our scenario into an image classification problem.

Figure 3 shows samples of the Lopez-Martinez dataset converted into image files that are used as input for the convolutional neural network. Each row of data contains 49 columns which yields a $7 \times 7 \times 1$ image. The ‘1’ means that the image is in grey scale.
We do not have to perform feature selection for convolution neural networks because that is done by the neural network itself through the convolution and max pooling layer. This is extremely useful as it makes use of all the features of a dataset. Hence, no feature is made redundant prior to max pooling. This is completely opposite to the artificial neural network; in which we have to perform feature selection in order to pick the best features based on strength score to be able to get the best results. Additionally, CNNs require fewer hyper-parameters and less supervision which gives it an advantage over other classification models.

The structure of the CNN allows it to learn the position and scale of features in a variety of images which makes the model very good at classifying hierarchical or spatial data, and extracting unlabelled features. However, the drawback is that the CNN can accept and provide only fixed-size inputs and outputs respectively.

**Recurrent neural network / long short-term memory (LSTM)**

Recurrent neural networks (RNN) are multi-layer neural networks that can analyse sequential, and time series input such as text, speech, videos and other forms of continuous data for classification and prediction purposes. Recurrent neural network is special in the sense that unlike artificial or convolutional neural networks, RNNs are not restricted to only feed forward movement. They perform by evaluating portions of an input and compare it with the portions both before and after it. This is done by the recurrent neural network by using weighted memory and feedback loops. The advantage of recurrent neural networks...
is that they are not limited by input length and can better predict meaning by using temporal context.

In RNNs, we obtain an output after each of our input is evaluated on a single layer. This can occur as one-to-one, one-to-many, many-to-one or many-to-many input to output basis. After the recurrent neural net evaluates the sequential features of the input, it returns an output to the evaluation step in a feedback loop. This allows the model to analyse the current feature in the context of the previous features.

When we train the recurrent neural network, we teach it to assign a suitable weight to each input feature. Afterwards, the RNN is able to determine which information is required to be sent back to the feedback loop with respect to gradient descent. This is known as Backpropagation Through Time (BPTT) as it creates a form ‘short-term memory’ for the recurrent neural net to refer back to.

However, the simple RNN has a vanishing gradient problem which means that the weights of the network sometimes receive such a small change that it appears to not have changed at all. In the worst-case scenario, this may prevent the model from training or training any further. This problem is solved by using Long Short-Term Memory which uses cell, an input and output gate, and a forget gate to remedy the vanishing gradient problem (Sak et al., 2014).

For our work, we have designed a recurrent neural network with two LSTM units of size 256 for the first unit (receives the input data), and 128 for the second unit. Two layers are reasonable for most scenarios, although the best number of layers vary on a case by case basis. More complex scenarios may benefit from more layers, however, that would make the model more difficult to train. We follow these two LSTM units with a feed forward fully-connected dense layer with 32 nodes. The two LSTM layers and the dense layer has ReLU activation. The LSTM layers have an additional recurrent activation function that is required to activate the input, forget, and output gate. We used sigmoid function for our recurrent activation. Additionally, each of the mentioned layers is followed by a dropout of 0.2 to prevent overfitting of data. The output has two nodes with ‘softmax’ activation similar to our other neural networks.

In the RNN, we have used Adam as the optimizer with a learning rate of 0.001 and decay rate of 1e-6. We use categorical cross-entropy to calculate the loss. Table 2 gives a list of hyper-parameters used to build the LSTM-RNN.

We input data into the RNN similar to how we entered input data into our convolution neural network. We pass the input as a two-dimensional matrix instead of three-dimensional. This is because as we are not using RNN as an image classifier, we do not have to convert our data into images to feed them to the network. Hence, the colour channel is not required for the matrix.

Figure 4 shows the model structure for our (a) CNN model and (b) LSTM-RNN with the corresponding hyper-parameters for each.

**Data source**

For our work, we use two datasets from two hospital in order to train and test our models. The datasets are from two different studies, and this gives us a variety in terms of data to work with. In this section, we will describe two datasets and discuss how we pre-processed these datasets to train with our models.
Lopez-Martinez published this dataset in (López-Martínez et al., 2019) and in this study we used it to train his artificial neural network. In the dataset, there are a total of 555 samples with two distinct classes. Out of the total number of samples, 34% is sepsis positive (186 samples) and 66% is sepsis negative (369 samples). The dataset is imbalanced with a positive to negative case ratio of 1:2. These sample data are collected from year 2016–2017 by the Crecer’s Hospital Centre in Cartagena, Colombia. Neonates under the age of 72 h were diagnosed with sepsis by clinical analysis and blood culture tests. Neonates that were deemed healthy by clinical criteria before and after the 72-hour mark are the control samples for the dataset. Figure 5 shows the class distribution of the dataset.

There are 46 independent variables present in the dataset. In our previous study (Alvi et al., 2020), we conducted feature selection on the dataset in order to select 27 independent variables that had the strongest effect on the dependent variable in terms of classification. However, in this study we opted to use all 46 independent variables in order to check how well our convolution and recurrent neural networks handle them. Hence, we do not require the strength score of individual variables unlike in the previous study. Additionally, since we would be passing the data as a matrix to our neural networks, having enough features to form a square matrix is more desirable than forming a $3 \times 9$ matrix with 27 features. This is because the resulting matrix is more skewed towards one dimension and is undesirable.

Moreover, in our previous study, we perform Random Oversampling and Undersampling in order to balance the dataset. After conducting Oversampling, the number of samples in the dataset increased to 738, with both the classes yielding 369 occurrences. Conversely, Undersampling decreased the total sample count to 372, with 186 samples getting lost from the majority class (control). Thus, we previously chose Oversampling over Undersampling for this dataset.

The drawback of using these balancing techniques is that it made the performance of the model slightly worse than normal. This is understandable because, even though Oversampling can be performed more than once (Ling & Li, 1998), duplicating

| Parameter                        | Value                             |
|----------------------------------|-----------------------------------|
| Input Shape                      | $7 \times 7$                      |
| Output class                     | 2                                 |
| Number of LSTM Layers            | 2                                 |
| LSTM Layer 1 Dimensions          | 256                               |
| Activation function for layer 1  | ReLU                              |
| Recurrent Activation function for layer 1 | Sigmoid                   |
| LSTM Layer 2 Dimensions          | 128                               |
| Activation function for layer 2  | ReLU                              |
| Recurrent Activation function for layer 1 | Sigmoid                   |
| Dense Layer Dimensions           | 32                                |
| Activation function for Dense layer | ReLU                        |
| Dropout                          | 0.2                               |
| Optimizer                        | Adam                              |
| Learning rate                    | 0.001                             |
| Loss function                    | Cross entropy with softmax        |
| Error Type                       | Classification error              |
| Parameter Learner                | Stochastic gradient descent       |
| Evaluation metrics               | Confusion Matrix, AUC             |

### Lopez-Martinez (DATASET 1)

Lopez-Martinez published this dataset in (López-Martínez et al., 2019) and in this study we used it to train his artificial neural network. In the dataset, there are a total of 555 samples with two distinct classes. Out of the total number of samples, 34% is sepsis positive (186 samples) and 66% is sepsis negative (369 samples). The dataset is imbalanced with a positive to negative case ratio of 1:2. These sample data are collected from year 2016–2017 by the Crecer’s Hospital Centre in Cartagena, Colombia. Neonates under the age of 72 h were diagnosed with sepsis by clinical analysis and blood culture tests. Neonates that were deemed healthy by clinical criteria before and after the 72-hour mark are the control samples for the dataset. Figure 5 shows the class distribution of the dataset.

There are 46 independent variables present in the dataset. In our previous study (Alvi et al., 2020), we conducted feature selection on the dataset in order to select 27 independent variables that had the strongest effect on the dependent variable in terms of classification. However, in this study we opted to use all 46 independent variables in order to check how well our convolution and recurrent neural networks handle them. Hence, we do not require the strength score of individual variables unlike in the previous study. Additionally, since we would be passing the data as a matrix to our neural networks, having enough features to form a square matrix is more desirable than forming a $3 \times 9$ matrix with 27 features. This is because the resulting matrix is more skewed towards one dimension and is undesirable.

Moreover, in our previous study, we perform Random Oversampling and Undersampling in order to balance the dataset. After conducting Oversampling, the number of samples in the dataset increased to 738, with both the classes yielding 369 occurrences. Conversely, Undersampling decreased the total sample count to 372, with 186 samples getting lost from the majority class (control). Thus, we previously chose Oversampling over Undersampling for this dataset.

The drawback of using these balancing techniques is that it made the performance of the model slightly worse than normal. This is understandable because, even though Oversampling can be performed more than once (Ling & Li, 1998), duplicating
certain samples caused the model to overfit. On the other hand, randomly removing certain samples in Undersampling increased the risk of high variance for the classifier (Fernández et al., 2018). To tackle these drawbacks, we decided against balancing the dataset.

Figure 6 shows a correlation matrix heat map for all 46 independent variables plus the dependent variable. It shows how strongly each variable is related to one another. Green represents strong relation whereas red represents weak relation.
By performing dimensionality reduction, we are able to plot all 46 variables into a 2D plane (Van Der Maaten et al., 2009; Van Der Maaten & Hinton, 2008). Figure 8 shows the 27 variables used in the previous study, whereas Figure 7 shows the 46 variables we used for this study. From the figures, we can clearly see that Figure 8 has well-established regions for both the classes. On the other hand, the same cannot be said after Figure 7, which does not have a clearly defined region for each class. This shows how feature selection can help make it easier for classifiers to distinguish distinct cases. However, since the CNN and LSTM-RNN are able to make use of all 46 variables, feature selection is not necessary.

We do not need to perform feature selection on the dataset for our CNN and RNN model. However, since we have to pass our data input as a matrix, we need to have the suitable number of features (columns) whose total number must be a multiple of \( N \) and \( M \) in order to have an \( N \times M \) matrix where \( N > 0 \) and \( M > 0 \). Hence, we have to modify the dataset in such a way that it can be transformed into an \( N \times M \) matrix. The following modification is done for only the CNN and RNN models, and it is not applicable for our artificial neural network or other machine learning classifiers.

Since patient ID and Serial numbers are not useful for training and testing, and the Hypothermia feature of the dataset contains all 0 values, these three columns are dropped from the dataset. Afterwards, 4 new columns are added as features to the dataset (pixel_1, pixel_2, pixel_3, pixel_4). These columns are all filled with 0 value which ensures that the nature of the dataset do not change. This makes the dataset have 50 columns (including the label column). Thus, once we remove the label for training and testing, we are left with a total of 49 columns that can be converted to a \( 7 \times 7 \) matrix for our CNN and RNN to accept, where \( N = M = 7 \). This makes the dataset very similar to the MNIST dataset for handwritten images, which contains pixel values of over 60,000 images of size \( 28 \times 28 \) (MNIST handwritten digit database, 2020). This is a famous dataset that is used in training convolutional networks and thus, was our inspiration to transform the DATASET 1 in a similar fashion.

**Figure 5.** Sepsis class distribution.
The dataset used in this study is from the Children’s Hospital of Philadelphia Neonatal Intensive Care Unit (CHOP NICU) sepsis registry establish in 2014 (Masino et al., 2019). The dataset is similar to the DATASET 1 as it contains automatically populated data abstracted from the EHR. The dataset has 618 uniquely identified infants with 1188 sepsis evaluations and is divided into two data subsets: CPOnly (cultural positive sepsis case) and CP + Clinical (culture positive and clinically positive).

The study provides two datasets, one with 375 case samples, and one with 1100 control samples. Unfortunately, the control dataset samples do not contain labels for each row. Hence we cannot use it to train and test our models.

The case dataset contains 111 positive cases and 264 negative cases with 36 columns (including the label). Hence, for our CNN and RNN model to be able to use this tabular

**Figure 6.** Correlation Matrix heat map of all 46 variables.

**AJ Masino (DATASET 2)**

The dataset used in this study is from the Children’s Hospital of Philadelphia Neonatal Intensive Care Unit (CHOP NICU) sepsis registry establish in 2014 (Masino et al., 2019). The dataset is similar to the DATASET 1 as it contains automatically populated data abstracted from the EHR. The dataset has 618 uniquely identified infants with 1188 sepsis evaluations and is divided into two data subsets: CPOnly (cultural positive sepsis case) and CP + Clinical (culture positive and clinically positive).

The study provides two datasets, one with 375 case samples, and one with 1100 control samples. Unfortunately, the control dataset samples do not contain labels for each row. Hence we cannot use it to train and test our models.

The case dataset contains 111 positive cases and 264 negative cases with 36 columns (including the label). Hence, for our CNN and RNN model to be able to use this tabular
Figure 7. Full variable set visualization.

Figure 8. Feature-selection set visualization.
data, we again modify the dataset by adding an additional column (pixel_0) and fill it up with all 0 values. This allows us to transform the dataset into a 6 × 6 matrix once we remove the label column. Thus, it becomes suitable for use in our deep learning neural networks.

**Result analysis**

In this section, we evaluate the performance of each of our models, and discuss and compare the results of each of the models to determine which classification works best for our scenario.

Precision, recall and f-measure are well known for their ability to measure performance specially related to pattern recognition and information retrieval task (Evaluation, 2020).

We first look in to the performance of the traditional machine learning algorithms used to predict neonatal sepsis from our DATASET 1 (Lopez-Martinez). KNN, SVM, Random Forest and Logistic Regression models have all performed well in this dataset. We can see from Table 3 that these models have almost identical scores.

From the above table we can clearly see that the results of each of these models are almost identical. All of these models are able to outperform the Neural Network model from the baseline study by a good margin. Moreover, these models are able to almost match the results of our artificial neural network. This shows that selection of important features based on strength score in a dataset is very important for getting high performance accuracy.

Table 4 shows the results achieved in this study and the results of the previous model. We can observe from the table that our model significantly outperforms the other one.

Next we test our more advance neural nets, the convolutional neural network and the long short-term memory recurrent neural network. Convolution Neural Networks are immensely powerful and are the industry standard in image classification models. On the other hand, LSTM-RNNs are the go-to models for natural language processing (NLP). Image classification and NLP are very complex scenarios compared to classification problems with straight-forward structured tabular datasets. This leads us to believe that CNN and LSTM will give promising result for our dataset. Results for these models are given in Table 5.

We can observe from Tables 4 and 5 that precision, recall, F-measure and accuracy of each of our models is almost identical. The artificial neural network we developed in our

| Table 3. Evaluation of traditional machine learning algorithms. |
|---------------------------------|----------------|-----------------|----------------|----------------|
| **Model**                        | **Precision** | **Recall**      | **F-measure**  | **Accuracy**   |
| KNN                             | 0.960         | 0.934           | 0.950          | 0.960          |
| SVM (linear kernel)             | 0.966         | 0.934           | 0.950          | 0.964          |
| Random Forest                   | 0.97          | 0.970           | 0.970          | 0.970          |
| Logistic Regression (L.BFGS)    | 1.000         | 0.918           | 0.957          | 0.970          |

| Table 4. Evaluation of ANN. |
|---------------------------------|----------------|----------------|----------------|----------------|
| **Model**                        | **Precision (%)** | **Recall (%)** | **F-measure (%)** | **Accuracy (%)** | **AUC (%)** |
| ANN                             | 96.77           | 98.40          | 97.60          | 98.20          | 99.80       |
| Lopez-Martinez Neural Net       | 83.00           | 80.30          | 81.070         | 86.74          | 92.50       |
previous work handily outperformed the model developed in the Lopez-Martinez study. Moreover, the convolution and long short-term memory recurrent neural networks developed in this study not only matched our previous model in terms of performance, but even overtook it in some areas.

The CNN has identical precision and recall percentage, which means that like our previous ANN, it can predict both positive and negative cases equally well. The LSTM-RNN performed even better, reaching a perfect recall rate of 1.00 versus the rate of 0.9840 of the ANN. The difference in precision between the RNN and ANN is approximately 0.60, which is low enough that the difference can be considered negligible.

The results also indicate that, even when using non-time series data, the LSTM-RNN can overtake the multi-layer perceptron that is specifically designed for static data. With an F-measure of 98.04% and an accuracy of 99.40%, the LSTM-RNN can be considered better at predicting early-onset neonatal sepsis cases better than the ANN and the CNN who achieved an F-measure score of 97.60% and 95.37%, and accuracy of 98.20% and 97.21% respectively.

The artificial neural network from our previous study is able to achieve a higher AUC of 99.80% against the 99.15% of the LSTM-RNN. However, as with the precision values, the value is low enough to be considered identical (less than 0.8). This shows that the LSTM-RNN is able to perform equal to or better than the ANN even when using all 46 variables from the dataset, something that the ANN did not do. The ANN required strong features to score on par with the LSTM-RNN.

We can evaluate that the CNN just falls under the performance of the ANN and RNN. However, it still performs sufficiently better than the traditional machine learning algorithms (kNN, SVM, random forest, logistic regression). The CNN is able to easily outperform the ANN developed in the Lopez-Martinez study, even after using tabular data and all of the 46 variables, same as the LSTM-RNN.

From this, we can clearly see that the LSTM-RNN outperforms the CNN model as well as give better results than the ANN. The CNN, surprisingly, falls short when it comes to comparing with the ANN and RNN, and gives performance similar to the SVM and Logistic Regression model.

Figure 9 shows the accuracy and loss performance of the CNN while Figure 10 shows the accuracy and loss performance of the LSTM-RNN.

This might make it look like that the CNN is not a good replacement for the ANN. However, if we look at the parameters of each of the neural network models from Table 6, we can see something astonishing. The total parameters for each model is obtained through `model.summary()` function from the keras framework.

We can clearly see that the jump in total parameters from our convolutional neural network to our artificial neural network is over 2000%. The recurrent neural network has the highest amount of total parameters which can explain why it gave the best results out of the three neural networks. The significantly lower parameters of the CNN can indicate why it did not match the ANN and RNN in its evaluation. However, we designed a

| Model   | Precision (%) | Recall (%) | F-measure (%) | Accuracy (%) | AUC (%) |
|---------|---------------|------------|---------------|--------------|---------|
| CNN     | 95.37         | 95.37      | 95.37         | 97.21        | 97.41   |
| LSTM-RNN| 96.15         | 1.00       | 98.04         | 99.40        | 99.15   |

| Table 5. Evaluation of CNN and LSTM-RNN model. |
simple CNN with only one convolution layer and max pooling layer. The performance of the neural network will increase and may reach that of the RNN if a more sophisticated selection of hyper-parameters is introduced. Performance of CNN increases with the number of convolution and pooling layers, in addition to the increase in dense layer just like all neural networks. On the other hand, too many layers may lead to the model overfitting the data which may even decrease the overall performance of the model. Hence, careful consideration must be given to selecting the hyper-parameters for the CNN with suitable dropouts to ensure that overfitting is minimized to the utmost (Figure 11).

Finally, we move onto our second dataset (DATASET 2) from the work of AJ Masino. We did not implement any traditional machine learning algorithms for this dataset as these models are already implemented by Masino. Hence, our choice is to use our CNN and RNN models to test their prediction capability when it comes to the tabular DATASET 2. Table 7 shows the performance of the models used by Masino et al. (2019) as well as having the results of the neural networks we used.

We can clearly see that two neural networks outperform the machine learning models used by Masino in their dataset. This shows that our models are robust enough to work on multiple datasets with excellent evaluation scores.

From the graph above, we can see that LSTM-RNN performs the best out of the three models. It can be misleading at first to see that the CNN performing poorly compared to
the other models. However, we have to take into consideration the computational
demand of each model type.

From Figure 12 we can tell that our CNN model performs most efficiently, having the
least amount of parameters required to achieve its scores. Additionally, adding more con-
volution layers, pooling layers and dense layers with the necessary dropouts will increase
the performance of the CNN.

Ultimately, we can conclude through this work that LSTM-RNN with its backpropaga-
tion through time performs the best when it comes to dealing with tabular data.

**Conclusion**

After analysing the scores of both the models implemented in this study, and the artificial
neural network model from our previous work, we can see that our neural network
architectures (ANN, CNN and LSTM-RNN) have a higher evaluation score than the models proposed in earlier related work. We developed tree classifiers, logistic regression models and support vector machine to compare with the results of our neural networks. Even though these non-neural network models performed better than most of the classifiers and achieved similar AUC scores of around 95%, they still could not compete with the neural networks. Thus, we can conclude that the powerful tools inherent in neural networks, especially CNN and LSTM-RNN, ensured this significant improvement in the model evaluations.

Additionally, we see that even with a split of 0.7:0.3, the error loss between the train and test set is identical. From this we are able to confidently say that these models have little to zero overfitting. On the other hand, we see that our CNN and LSTM-RNN models can perform similarly in the Masino dataset as well. This shows that these models are not limited by only one dataset, and that they are able to perform when given different data as well. Three neural networks also have similar scores for sensitivity, precision and specificity. Thus, we can use these models to predict both positive and negative cases equally well. It is an important distinguisher for the models as the neural network model published by Lopez-Martinez was visibly better in detecting negative cases than it was in detecting positive cases.

We have previously mentioned; our models are suitable for implementation as part of an expert system in Neonatal Intensive Care Units (NICU) to administer correct treatments for sepsis (EOS). Moreover, our model is designed to give prediction as soon as data is provided; this can be 4 h or more before actual clinical suspicion. However, the model is not sophisticated enough to replace clinical experts.

The convolutional and recurrent neural networks used in this study are renowned and very powerful architectures in the field of deep learning. They are mostly (almost exclusively) used for continuous, time-series data that require powerful and complex architectures. However, in this study we applied these powerful models on static, tabular data (non-time series) which are considered less complex than time-series. The results are overwhelmingly positive as the best performing model for this dataset came out to be
the LSTM-RNN model which achieved an accuracy of 99.40%, the highest among all the models. We can observe from this that these powerful architectures are well suited for static, tabular data and can outperform even the artificial feedforward neural network.

Any form of research work will always have room for improvements and it holds true for our work as well. As mentioned earlier, the convolution neural networks hold the most room for improvement. The CNN model also shows the most promise as it is able to best all the traditional machine learning algorithms even after using 289,204 parameters less than an artificial neural network. The CNN used only one convolution layer and one max-pooling layer. Increasing the number of convolution and pooling layers have shown to effectively increase the performance of convolution layers. An example of such a CNN would be the GoogLeNet (Inception V1) which is a 22 layer deep CNN with 4 million parameters (Szegedy et al., 2014).

Our design proved that the powerful features of these convolutions and LSTM cells can be used to utilize all the data that is available, and not get limited by only sequential or time-series data, due to the data-hungry nature of the models.

There are, however, limitations to the networks described in this study. Since neural networks are data intensive architectures, and thrive with increasing data, more complex datasets with higher features can be included in the future. Reinforced learning may be another technique that can be used to see how well a model can perform on tabular data. Finally, transfer learning with CNN and RNN can be used to bring a much more sophisticated pre-trained model to be used to predict early onset neonatal sepsis.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Redwan Hasif Alvi will be completing his BSc. in Computer Science and Engineering from the Department of Electrical and Computer Engineering, North South University. His research interests are in the field of deep learning and reinforced learning, with particular interest in pattern recognition and neural networks. He has a history of publications in IEEE Intelligent System conference.

Md. Habibur Rahman will be completing his BSc. in Electrical and Electronic Engineering from North South University. His research interest includes artificial intelligence and robotics.

Adib Al Shaeed Khan is currently completing his B.Sc. in Electrical and Electronic Engineering from North South University.

Rashedur M. Rahman is working as a Professor in the Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. He received his PhD in Computer Science from University of Calgary, Canada. He has authored more than 150 peer-reviewed research papers. His current research interests are in data science, cloud load characterization, optimization of cloud resource placements, deep learning, etc. He is serving as an editorial board member in a number of renowned journals and a program committee member of several conferences organized by ACM, IEEE and Springer.

References

Adadelta. (2020). [Online]. Retrieved June 7, 2020, from https://keras.io/api/optimizers/adadelta/
Adatara, P., Afaya, A., Salia, S. M., Afaya, R. A., Konlan, K. D., Agyabeng-Fandoh, E., Agbinku, E., Ayandayo, E. A., Boahene, I. G. (2019). Risk factors associated with neonatal sepsis: A case study at a specialist hospital in Ghana. The Scientific World Journal, 2019, 1–8. https://doi.org/10.1155/2019/9369051

Alvi, R. H., Habibur Rahman, M., Al Shaeed Khan, A., & Rahman, R. M. (2020, 28–30 Aug). Predicting early neonatal sepsis using neural networks and other classifiers. 10th. IEEE Conference on Intelligence Systems, Sofia, Bulgaria, https://doi.org/10.1109/is48319.2020.9199937

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: A comparative review. Journal of Machine Learning Research 10, 66–71.

Calvert, J. S., Price, D. A., Chettipally, U. K., Barton, C. W., Feldman, M. D., Hoffman, J. L., Jay, M., Das, R. (2016). A computational approach to early sepsis detection. Computers in Biology and Medicine, 74, 69–73. https://doi.org/10.1016/j.compbiomed.2016.05.003

Desautels, T., Calvert, J., Hoffman, J., Jay, M., Kerem, Y., Shieh, L., Shimabukuro, D., Chettipally, U., Feldman, M. D, Barton, C., Wales, D. J, Das, R. (2016). Prediction of sepsis in the intensive care unit with minimal electronic health record data: A machine learning approach. JMIR Medical Informatics, 4(3), e28. https://doi.org/10.2196/medinform.5909

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61, 863–905, AI Access Foundation, 01-Apr-2018, https://doi.org/10.1613/jair.1.11192.

Gómez, J. L., González, S. C., Muñoz Baldiris, R. E., Díaz-Pérez, A., & Perez, I. (2018). Prognostic factors of early neonatal sepsis in the city of Cartagena Colombia. Global Journal of Health Science, 10(12), 30. https://doi.org/10.5539/gjhs.v10n12p30

Griffin, M. P., Lake, D. E., & Moorman, J. R. (2005). Heart rate characteristics and laboratory tests in neonatal sepsis. Pediatrics, 115(4), 937–941. https://doi.org/10.1542/peds.2004-1393

Horng, S., Sontag, D. A., Halpern, Y., Jernite, Y., Shapiro, N. I., Nathanson, L. A., & Groza, T. (2017). Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. PLoS One, 12(4), e0174708. https://doi.org/10.1371/journal.pone.0174708

MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. (2020). [Online]. Retrieved June 8, 2020, from https://yann.lecun.com/exdb/mnist/
Nagi, J., Ducatelle, F., Di Caro, G. A., Ciresan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella., L. M. (2011, 16–18 November). *Max-pooling convolutional neural networks for vision-based hand gesture recognition*. 2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011.

Nguyen, H. B., Corbett, S. W., Steele, R., Banta, J., Clark, R. T., Hayes, S. R., Edwards, J., Cho, T. W., Wittlake, W. A. (2007). Implementation of a bundle of quality indicators for the early management of severe sepsis and septic shock is associated with decreased mortality*.* Critical Care Medicine, 35(4), 1105–1112. https://doi.org/10.1097/01.CCM.0000259463.33848.3D

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12.

Sak, H., Senior, A., & Beaufays, F. (2014). *Long short–term memory recurrent neural network architectures for large scale acoustic modeling*. 15th Annual conference of the international speech communication association Singapore, September 14–18, 2014.

Singh, M., Alsaleem, M., Gray, C. P. (2020). Neonatal sepsis. In *StatPearls*. StatPearls Publishing.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). *Going deeper with convolutions*. Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 2015, vol. 07-12-June-2015, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Van Der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

Wu, H., & Gu, X. (2015). *Max-pooling dropout for regularization of convolutional neural networks*. Proceedings of the 22nd international conference neural information processing (pp. 46–53), 2015.