

Real Time Mortality Risk Prediction: A Convolutional Neural Network Approach

Landon Brand, Aditya Patel, Izzatbir Singh and Clayton Brand
Stasis Labs, Los Angeles, U.S.A.

Keywords: CNN, Machine Learning, Mimic, Mortality Prediction.

Abstract: Machine Learning in Healthcare shows great promise, but is often difficult to implement due to difficulties in collecting data. We used a 1-dimensional convolutional neural network(CNN) on limited data to show a practical application of deep learning in healthcare. We used only vital signs data that can be collected from low cost, readily available hardware designed for non-critical care settings, and a dynamic model that updates as more data is collected over time. Our data is derived from the MIMIC dataset. We use 320 patients for testing and 2,990 for training the model. The CNN model predicted mortalities with up to a 76.3% accuracy, and outperformed both recurrent neural network and multi-layer perceptron models. To our knowledge, the proposed methodology is the first of its kind to predict mortality risk scores based on only heart rate, respiratory rate, and blood pressure, three easily collectible data.

1 INTRODUCTION

1.1 Background on Severity Scores in Mortality Prediction

Electronic Medical Records (EMR) have been a rich source of patient care data for predictive risk assessment models. Utilizing this data could significantly improve severity of illness assessments and assists clinicians in deciding interventions for patients. Several scores have been devised and tested to predict Mortality risk using the first 24 hours of patient physiological measurements after ICU admission. Widely used severity scores in clinical practice are APACHE II (Knaus *et al.*, 1985) (Acute Physiology and Chronic Health Evaluation) and SAPS II (Le Gall, Lemeshow and Saulnier, 1993) (Simplified Acute Physiology Score). These scores have been modified over time to improve their predictive performance. The initial scores proposed—APACHE (Knaus *et al.*, 1981), APACHE II (Knaus *et al.*, 1985) and SAPS (Le Gall *et al.*, 1984)—relied on assigning weights to physiological measures, decided by a panel of experts, whereas SAPS II (Le Gall, Lemeshow and Saulnier, 1993) was obtained through Statistical modelling techniques. Studies have been conducted to validate (Knaus *et al.*, 1985) and compare (Nassar *et al.*, 2012; Poole *et al.*, 2012) the reliability of

these severity scores for predicting risk, but even after revisions the probability of mortality is overestimated by these scores (Nassar *et al.*, 2012; Poole *et al.*, 2012). Further modifications on severity scores included APACHE IV and SAPS 3 scores, but their performance as evaluated in (Nassar *et al.*, 2012) was no better than the existing scores.

Most of these scores are based on logistic regression models. Logistic regression models run on strict assumptions on dependent and independent variables which may not be always true. For instance, some interventions may impact patients in a non-linear way. Non-parametric methods have been used to overcome the constraints of logistic regression models. Studies have shown that non-parametric methods based on neural networks can perform at least to the baseline models given by logistic regression models (Dybowski *et al.*, 1996; Clermont *et al.*, 2001; Foltran *et al.*, 2010; Kim, Kim and Park, 2011; Ribas *et al.*, 2011).

1.2 Background on Machine Learning in Healthcare

In statistics and healthcare, Autoregressive Integrated Moving Average (ARIMA) models are widely used in time series forecasting. These models don't assume any prior knowledge about underlying model and depends only on past data and error

values which makes them more robust and easy to explain. But these models are generated based on an assumption that time series are generated from linear processes which makes it inappropriate for real world problems. On the contrary, deep learning methods are self-adaptive with few prior assumptions. They are able to generalise the learnings from original data and are good for solving non-linear problems. Hence, ARIMA models are out of scope for this paper. There exists literature comparing traditional moving average models and Neural networks for time series forecasting (Guoqiang Zhang, B.Eddy Patuwo and Michael Y. Hu, 1998; Adebisi, Adewumi and Ayo, 2014).

Machine learning has been successfully used for many health-care related tasks, such as arrhythmia detection (Rajpurkar *et al.*, 2017), clinical intervention prediction (Suresh *et al.*, 2017), ICU transfer prediction (Yoon *et al.*, 2016) and more. These models have proved to perform up to the benchmarks and even beat the benchmarks in some cases. Regression based algorithms, Super learners were developed to choose an optimal regression model from a given set of models (Dudoit and van der Laan, 2005) for improving severity scores. Reference (Pirrachio, 2016) gives a severity score based on a super learner validated on the MIMIC II data set.

EMR data makes many machine learning prediction tools possible. It has encouraged the use of complex algorithms like Artificial Neural Networks and Decision Trees in healthcare problems. These new modelling approaches lead to many predictive models for different critical care settings (Ganzert *et al.*, 2002; Moser, Jones and Brossette, 1999; Morik *et al.*, 2000; Sierra *et al.*, 2001; Kong, Milbrandt and Weissfeld, 2004; Kreke, Schaefer and Roberts, 2004; Lucas, 2004). Reference (Kim, Kim and Park, 2011) compares results of machine learning techniques for Mortality prediction models inside ICU.

Deep Neural Networks (DNNs) have been shown to be effective in predicting mortality in paediatric healthcare settings (Aczon, Ledbetter *et al.*, 2017) (Nguyen, Tran and Venkatesh, 2017). They are especially useful when dealing with high dimensional data, which is common in healthcare. Though Recurrent Neural Networks (RNNs) are used more often for time-series data, recently Convolutional Neural Networks (CNNs) have also been used with medical time series data to achieve state of the art results (Suresh *et al.*, 2017).

1.3 Limits on Available Data

Though it is easy to develop machine learning models that can work after a patient has left the hospital, in non-critical care settings most data is inputted into EMR in an untimely manner after it is recorded (McGain *et al.*, 2008). However, there are products that will monitor patients continuously and automatically send the data to a central server, specifically designed for non-critical care settings. Over the past 15 years, numerous vital signs monitoring systems have been developed for non-critical care settings (Patel *et al.*, 2012). These monitoring systems might not capture all the information that is captured inside the ICU, but they do capture body vital signs.

On sharing of health care data, The Health Information Technology for Economic and Clinical Health (HITECH) Act (*HITECH Act Enforcement Interim Final Rule | HHS.gov*, 2009) promotes the meaningful use of EMR data. But this data comes with privacy concerns. There are strict laws governing the use of this data. The Health Insurance Portability and Accountability Act (HIPAA) regulations apply on all the health care providers to protect the patient information. Patients are given rights to control their medical data under the Act. All the data is owned by the patients and a written consent is required to use it. Any violation of the Act will attract both civil and criminal penalties. This makes it difficult for researchers and practitioners to make use or share the health care data for experiments.

The potential impact of Machine Learning in healthcare is large, however it is difficult to implement machine learning solutions in real clinical settings. To be deployed to real settings, machine learning algorithms must use data that is accessible. For algorithms that rely on making predictions using extensive accurate, time-sensitive data, this is problematic. In non-critical care settings patient monitoring is not continuous. The data collected in such settings may be incomplete and may take hours before being reported to the system. Hence the above-mentioned machine learning models might not perform as desired because of incomplete data. To address this, in this paper we discuss a model using a CNN which predicts mortality risk in settings where access to data is limited.

2 METHODS

2.1 Data Source

We use data from the Medical Information Mart for Intensive Care (MIMIC-III) database (Johnson et al., 2016). MIMIC-III is a deidentified publicly available dataset that contains detailed health records from approximately 40,000 critical care patients, including vital sign recordings. We consider only patients that have at least 12 hours of records where each hour has at least 1 recording for heart rate, respiratory rate, systolic blood pressure, and diastolic blood pressure. There are 1,814 of these patients who passed away during their hospital stay, which we consider to be the mortality class. The number of patients who have the requisite vital sign recordings but survive their stay in the hospital is larger, but we take only the first 1,814 of these patients, as ordered by the MIMIC database, into consideration so that the 2 classes are equal size. This leads us to use 3,628 patients in total, exactly half of which are survivors and the other half of which are mortalities.

2.2 Data Preparation

To simulate non-critical care settings, we consider only physiological vital sign data to make our predictions. For each patient, we construct a 5 x 47 matrix where each row contains 47 readings corresponding to each hour of their stay. The 4 vital signs we use are heart rate, respiratory rate, systolic blood pressure, and diastolic blood pressure. The data is normalized per feature, such that the mean of each vital sign is 0, and the variance of each vital sign is 1. We then simulate the patient's stay by creating a matrix that resembles the available data for each hour of their stay, for a total of up to 47 separate matrices for each patient used as input to our model. This way we limit the threshold of data collected to 47 hours for each patient to make the prediction. We also add an additional row of the matrix to denote whether there is data yet recorded for each hour. This row is 0 for time values where no vital signs are yet recorded, and 1 for time values where vital signs are present. To make it real time and adaptive to variable time lengths, we increase the sample size by assuming a new sample for each hour of patient vitals recorded. This simulates a patient's stay in a non-critical care setting (Figure 1). Each hour, their vital signs are recorded, and the

model can be re-run using this new data for a more accurate estimate.

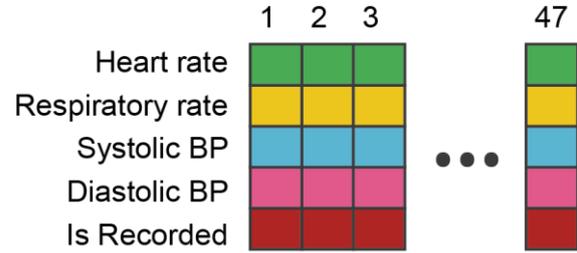


Figure 1: Diagram of the final input to the model after data preparation.

In the MIMIC dataset, some patients have more vital sign recordings per hour than others. In this case, we simply take the last vital sign recorded in that hour as the value to go in the matrix.

At this point, there are 170,516 samples that correspond to a specific hour of a patient's stay. We then take out 15,000 of these samples to use in the validation data set, and another 15,000 of these samples to use in the testing data set. We use the validation data set to tune our hyperparameters for the models, and the testing data when evaluating the models in section 3. The remaining 140,516 samples are used as the training data

2.3 Convolutional Neural Networks (CNNs)

2.3.1 Basic Structure

Neural Networks employ the back-propagation algorithm to calculate optimum weights to predict the output class (Equation 1). The output of a single neuron is defined as a_j^l of j^{th} row and l^{th} layer, where w^l are weights connected to the l^{th} layer of neurons and sum is over all k neurons in the $(l-1)^{\text{th}}$ layer. The aim is to find weights, which ensure that for each input the output produced by the network is the same as the desired output vector. To minimise the error, gradient methods are used, and the errors are back propagated through chain rule. A comprehensive review is provided in (Lecun *et al.*, 1998; Krizhevsky, Sutskever and Hinton, 2012a)

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right) \quad (1)$$

Convolutional neural networks (CNN) stand out as an example of neuroscientific principles influencing neural network architecture (Goodfellow, Bengio

and Courville, 2016). CNN Models enable us to exploit a multiple layer architecture for non-linear information processing, to extract features for classification tasks (LeCun *et al.*, 1989). CNN architectures come in various forms but usually they consist of modules, which have a convolution, and pooling (subsampling) layer. These modules are stacked on top of each other to create deep learning models. The last of these modules are connected to a fully connected feed forward neural network to do the classification tasks. As given in (Lecun *et al.*, 1998) Figure. 2 illustrates a typical CNN Model.

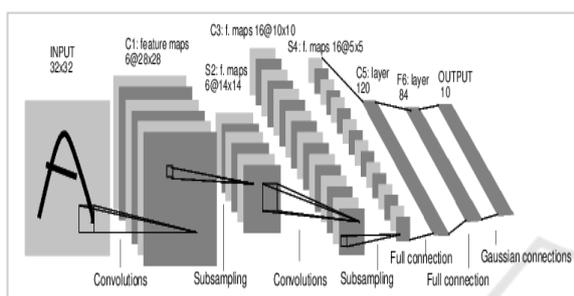


Figure 2: Illustration of typical CNN Architecture.

In deep CNN layers, units in the deeper layer can indirectly interact with large portions of the input which lead it to learn the underlying structure/patterns of the input data without using hand designed features. This enables us to find features which we might miss out on using other types of models. Mathematically convolution can be defined by (Equation 2), where $s(t)$ is the output, $x(a)$ is the input weighted by the weighting function $w(a)$. Convolution in CNN's are defined by (Equation 3) where, $h_j^{(m)}$ is the output of the m^{th} layer with weights w , input v , bias a_j and θ the activation function. The function the layer learns contains local interactions and is equivariant.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (2)$$

$$h_j^{(m)} = \theta \left(\sum_{b=1}^s w_{b,j}^{(m)} v_{(m-1)*n+b}^T + a_j^{(m)} \right) \quad (3)$$

Usually pooling layers provide summary statistics of nearby outputs in the feature map. The use of pooling layers enables us to make the representation become invariant to small translation of the input. The applications of CNNs are widely in image processing domain, but the CNN architecture can

also be applied in time domain to make sense of temporal data.

2.3.2 Model Architecture

In our current architecture, we apply CNN on time domain on multivariate time series. The inspiration to use a CNN to predict mortality is attributed to the sparse interaction in the feature map, weight sharing, and equivariance CNNs offer. The sparse interaction enables us to process the input quickly in real time, weight sharing aids in finding patterns along the time axis and equivariance helps in handling the input changes which are carried forward in the output (Goodfellow, Bengio and Courville, 2016). Figure. 3 gives a high-level architecture of the proposed methodology. The network takes the normalised input time series (described above) as input and model outputs a mortality risk score.

Processing Block: A Processing Block comprises of a convolution layer, activation function and a dropout layer.

Convolution Layer: The convolution layers have equal filter sizes with variable kernel sizes and strides (Equation 4). Convolution is done only along the time dimension of the input vectors giving us a 1-D CNN. The convolution layer input is padded to keep the output the same size as the input.

$$Z_{i,j,k} = c(K, V, s)_{i,j} = \sum_{l,m} [V_{l,(j-1)*s} K_{i,l,m}] \quad (4)$$

In (Equation 4) Z (output) is of the same format as V (input) and each value is addressed within row j and channel i . K gives the connection strength between Z and V . s is the stride which can down sample by skipping over some positions to reduce computational cost (Goodfellow, Bengio and Courville, 2016).

Activation Layer: In the current architecture, the Rectified Linear Unit (ReLU) function (Equation 5) is used to transform the feature map non-linearly. It calculates:

$$f(x) = \max(0, x) \quad (5)$$

ReLU have been found to greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid or tanh functions (Krizhevsky, Sutskever and Hinton, 2012b). We apply ReLU in conjunction with the convolution layer.

Dropout Layer: To reduce overfitting, we employ a dropout layer in the processing block. Dropout (Srivastava *et al.*, 2014) is a technique where we ignore randomly selected neurons from a layer during training. Essentially their (dropped out neurons) contributions to the activation of neurons downstream are removed during the forward pass and weight updates are not applied during the backward pass. This results in network which can generalize better. After calculating feature maps over multiple processing blocks, the feature map is flattened to connect it to a fully connected layer with a SoftMax activation function ($f_i(j)$) (Equation 7). This lets us calculate the probability of patient mortality. We optimise the cross-entropy objective function (L_i) for a single example in the training set (Equation 6).

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{6}$$

$$\text{Where, } f_i(j) = \left(\frac{e^{z_j}}{\sum_k e^{z_k}}\right) \tag{7}$$

In our proposed architecture, we have actively avoided pooling layers in the processing blocks (Figure 3). One of the main reason for us to avoid pooling layer is the low sampling rate of the given physiological data. Pooling layers would abstract away from the nuance changes which are needed to distinguish between various effects in training the classifier effectively.

Table 1: The hyperparameters used in each architecture. These hyperparameters were tuned by hand by trying many architectures and selecting the best one for each type of model.

| Model | Layers | # Hidden Units |
|-------|---------------|----------------|
| CNN | 4 convolution | 128 |
| RNN | 2 LSTM | 64 |
| MLP | 4 dense | 128 |

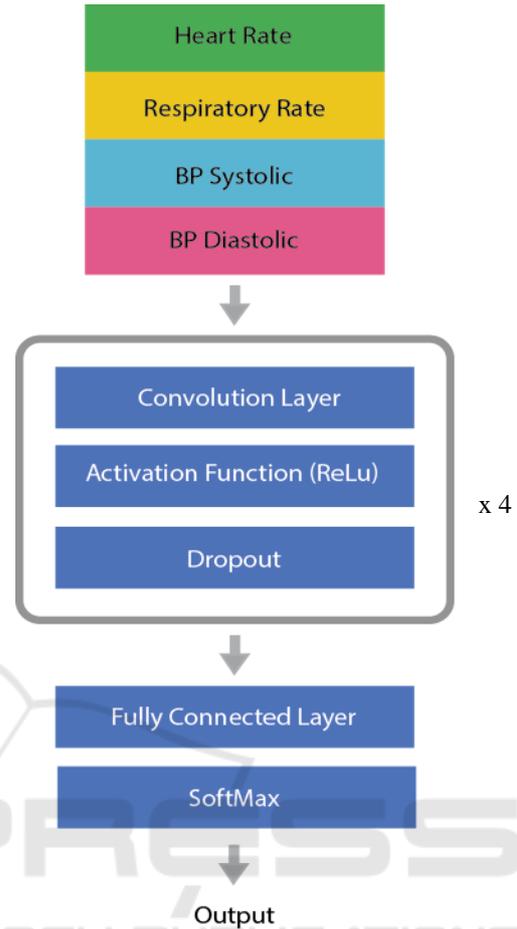


Figure 3: The architecture of the network. Overall it contains 4 layers followed by fully connected layer and SoftMax.

3 RESULTS

We compare the one-dimensional CNN model with two other deep learning models: A multilayer perceptron (MLP) model, and a recurrent Long Short-Term Memory (RNN) model (Hochreiter S. Schmidhuber J., 1997). The hyperparameters used are shown in Table 1. As figure [4] shows, the CNN outperforms both models at most points in time. For some cases around hour 20, 25, and 41, the RNN model outperforms the CNN. We achieve an accuracy of 0.76 with the CNN model predicting based off 47 hours of data, compared to an accuracy of 0.71 with the MLP model and 0.72 with the RNN model. As the Figure shows, these results are somewhat sporadic—in some cases one model performs better than another. In general, the CNN is the most successful.

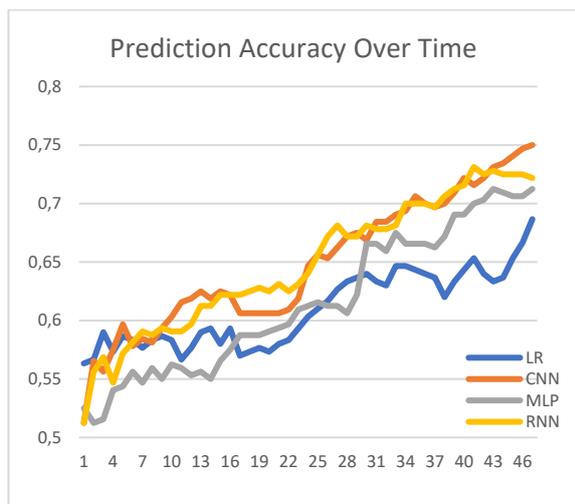


Figure 4: Accuracy comparison of various models over time (hours).

We also include a comparison with a Logistic Regression(LR) model using features selected by hand, specifically the maximum, minimum, mean, and standard deviation of each vital sign. The LR model is successful in predicting based on only a few hours of data, but LR fails to make sense of increasing data effectively, leading to an average accuracy of 0.61 and a peak accuracy of 0.69. This makes sense, as LR has no way of working with time as a dimension, but the deep learning methods can incorporate temporal position in their models effectively.

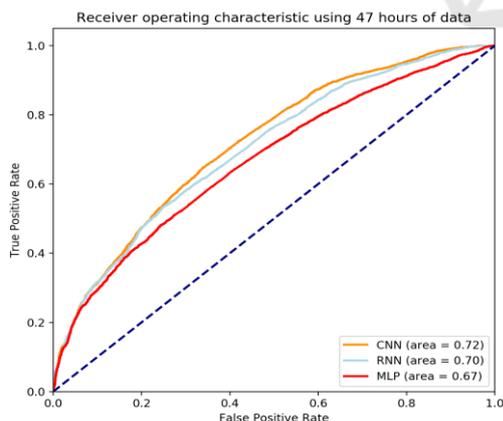


Figure 5: ROC comparing deep learning models.

Figure [5] shows the Receiver Operating Characteristic (ROC) curve (Hanley J McNeil B, 1982) of the three deep learning models. We used the entire set of results from all hours for this figure. The CNN model has the highest Area Under the

Curve (AUC), with 0.72. The RNN Model has an only slightly lower AUC at 0.70, followed by the MLP model with an AUC of 0.67.

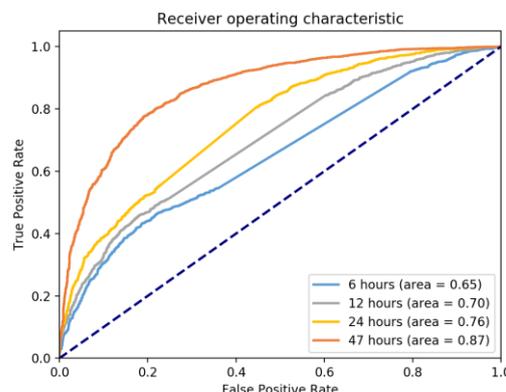


Figure 6: ROC comparing the CNN model using different time frames of data as input.

Figure [6] shows the ROC curve using different amounts of data in the CNN model. The blue curve uses only six hours of data, and achieves a meagre AUC of 0.65. As more hours of data are used, the CNN model’s accuracy clearly improves, until with 47 hours of data the CNN model achieves an AUC of 0.87.

Table 2: The Area under receiver operating characteristic, mortality class precision, and mortality class recall for each model when using 47 hours of data. The largest number in each column is bolded.

| Model | AUC | Precision | Recall |
|---------------------|-------------|---------------|---------------|
| CNN | 0.87 | 0.7443 | 0.8188 |
| RNN | 0.80 | 0.6940 | 0.7987 |
| MLP | 0.84 | 0.8091 | 0.5597 |
| Logistic Regression | 0.79 | 0.8069 | 0.5138 |

4 CONCLUSIONS

We find 1-dimensional CNNs to be a promising model for predicting mortalities using variable length vital signs data. Using this model, we can assess patient risk each hour using minimal equipment, as only low-frequency vital signs are needed, and have patient risk scores that update automatically over time. The CNN can predict with 60.3% accuracy after 12 hours of data collection, and 76.3% after 47 hours of data collection, and on average outperforms both an RNN model and a MLP model. A system like this could be helpful in

non-critical care settings, where patients may quietly deteriorate.

5 FUTURE WORK

Though this system currently has narrowly limited usefulness, because patients in non-critical care settings rarely pass away, in future work we intend to assemble a suite of risk assessment tools for patients in non-critical care settings based on the data that is unique to that setting. In the case that a patient in a non-critical care setting has a high predicted probability of mortality, this is a highly preventable death, and hospitals should be aware of the patient's condition and act accordingly. In the future, we would like to experiment with higher frequency sample data, to understand its impact on the results.

We also want to experiment with Deconvolution networks to understand the behaviour of various layers activations (Zeiler *et al.*, 2010). Deconvolution networks typically consists of unpooling and transposed convolution layers, the maximum activation of feature maps from each layer is passed through the earlier layers to reconstruct the inputs. This reconstructed input can give us an inkling on the patterns which can lead to mortality. It would be interesting to understand the activation patterns for patients with different conditions (Wang *et al.*, 2016). This can lead to building diagnostic tools which can be implemented on the monitoring devices to classify conditions of patient deterioration in real time.

REFERENCES

- Adebiyi, A. A., Adewumi, A. O. and Ayo, C. K. (2014) 'Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction', *Journal of Applied Mathematics*. Hindawi, 2014, pp. 1–7. doi: 10.1155/2014/614342.
- Clermont, G. et al. (2001) 'Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic regression models.', *Critical care medicine*, 29(2), pp. 291–6. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11246308> (Accessed: 31 August 2017).
- Dudoit, S. and van der Laan, M. J. (2005) 'Asymptotics of cross-validated risk estimation in estimator selection and performance assessment', *Statistical Methodology*, 2(2), pp. 131–154. doi: 10.1016/j.stamet.2005.02.003.
- Dybowski, R. et al. (1996) 'Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm.', *Lancet* (London, England). Elsevier, 347(9009), pp. 1146–50. doi: 10.5555/URI:PII:S0140673696906091.
- Foltran, F. et al. (2010) 'Using VLAD scores to have a look insight ICU performance: towards a modelling of the errors', *Journal of Evaluation in Clinical Practice*, 16(5), pp. 968–975. doi: 10.1111/j.1365-2753.2009.01240.x.
- Le Gall, J. R. et al. (1984) 'A simplified acute physiology score for ICU patients.', *Critical care medicine*, 12(11), pp. 975–7. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/6499483> (Accessed: 31 August 2017).
- Le Gall, J. R., Lemeshow, S. and Saulnier, F. (no date) 'A new Simplified Acute Physiology Score (SAPS II) based on a European/North American multicenter study.', *JAMA*, 270(24), pp. 2957–63. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/8254858> (Accessed: 31 August 2017).
- Ganzert, S. et al. (no date) 'Analysis of respiratory pressure-volume curves in intensive care medicine using inductive machine learning.', *Artificial intelligence in medicine*, 26(1–2), pp. 69–86. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12234718> (Accessed: 31 August 2017).
- Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press. Available at: <http://www.deeplearningbook.org/> (Accessed: 1 September 2017).
- Guoqiang Zhang, B.Eddy Patuwo and Michael Y. Hu (1998) 'Forecasting with artificial neural networks: The state of the art', *International Journal of Forecasting*. Elsevier, 14(1), pp. 35–62. doi: 10.1016/S0169-2070(97)00044-7.
- HITECH Act Enforcement Interim Final Rule | HHS.gov (no date). Available at: <https://www.hhs.gov/hipaa/for-professionals/special-topics/HITECH-act-enforcement-interim-final-rule/index.html> (Accessed: 26 October 2017).
- Kim, S., Kim, W. and Park, R. W. (2011) 'A Comparison of Intensive Care Unit Mortality Prediction Models through the Use of Data Mining Techniques.', *Healthcare informatics research*. Korean Society of Medical Informatics, 17(4), pp. 232–43. doi: 10.4258/hir.2011.17.4.232.
- Knaus, W. A. et al. (1981) 'APACHE-acute physiology and chronic health evaluation: a physiologically based classification system.', *Critical care medicine*, 9(8), pp. 591–7. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/7261642> (Accessed: 31 August 2017).
- Knaus, W. A. et al. (1985) 'APACHE II: a severity of disease classification system.', *Critical care medicine*, 13(10), pp. 818–29. Available at:

- <http://www.ncbi.nlm.nih.gov/pubmed/3928249>
(Accessed: 29 August 2017).
- Kong, L., Milbrandt, E. B. and Weissfeld, L. A. (2004) 'Advances in statistical methodology and their application in critical care.', *Current opinion in critical care*, 10(5), pp. 391–4. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15385757> (Accessed: 31 August 2017).
- Kreke, J. E., Schaefer, A. J. and Roberts, M. S. (2004) 'Simulation and critical care modeling.', *Current opinion in critical care*, 10(5), pp. 395–8. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15385758> (Accessed: 31 August 2017).
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012a) 'ImageNet Classification with Deep Convolutional Neural Networks', pp. 1097–1105. Available at: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks> (Accessed: 29 August 2017).
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012b) 'ImageNet Classification with Deep Convolutional Neural Networks', pp. 1097–1105. Available at: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks> (Accessed: 31 August 2017).
- Lecun, Y. et al. (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, 86(11), pp. 2278–2324. doi: 10.1109/5.726791.
- LeCun, Y. et al. (1989) 'Backpropagation Applied to Handwritten Zip Code Recognition', *Neural Computation*, 1(4), pp. 541–551. doi: 10.1162/neco.1989.1.4.541.
- Lucas, P. (2004) 'Bayesian analysis, pattern analysis, and data mining in health care.', *Current opinion in critical care*, 10(5), pp. 399–403. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/15385759> (Accessed: 31 August 2017).
- McGain, F. et al. (2008) 'Documentation of clinical review and vital signs after major surgery', *Med J Aust*. Available at: https://www.mja.com.au/system/files/issues/189_07_061008/mcg11494_fm.pdf (Accessed: 28 July 2017).
- Morik, K. et al. (2000) 'Knowledge discovery and knowledge validation in intensive care.', *Artificial intelligence in medicine*, 19(3), pp. 225–49. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10906614> (Accessed: 31 August 2017).
- Moser, S., Jones, W. T. and Brossette, S. E. (1999) 'Application of Data Mining to Intensive Care Unit Microbiologic Data', *Emerging Infectious Diseases*, 5(3), pp. 454–457. doi: 10.3201/eid0503.990320.
- Nassar, A. P. et al. (2012) 'Caution when using prognostic models: A prospective comparison of 3 recent prognostic models', *Journal of Critical Care*, 27(4), p. 423.e1–423.e7. doi: 10.1016/j.jcrc.2011.08.016.
- Patel, S. et al. (2012) 'A review of wearable sensors and systems with application in rehabilitation', *Journal of NeuroEngineering and Rehabilitation*. *BioMed Central*, 9(1), p. 21. doi: 10.1186/1743-0003-9-21.
- Pirracchio, R. (2016) 'Mortality Prediction in the ICU Based on MIMIC-II Results from the Super ICU Learner Algorithm (SICULA) Project', in *Secondary Analysis of Electronic Health Records*. Cham: Springer International Publishing, pp. 295–313. doi: 10.1007/978-3-319-43742-2_20.
- Poole, D. et al. (2012) 'Comparison between SAPS II and SAPS 3 in predicting hospital mortality in a cohort of 103 Italian ICUs. Is new always better?', *Intensive Care Medicine*, 38(8), pp. 1280–1288. doi: 10.1007/s00134-012-2578-0.
- Rajpurkar, P. et al. (2017) 'Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks'. Available at: <http://arxiv.org/abs/1707.01836>.
- Ribas, V. J. et al. (2011) 'Severe sepsis mortality prediction with relevance vector machines', in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, pp. 100–103. doi: 10.1109/IEMBS.2011.6089906.
- Sierra, B. et al. (2001) 'Using Bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patients data', *Artificial Intelligence in Medicine*, 22(3), pp. 233–248. doi: 10.1016/S0933-3657(00)00111-1.
- Srivastava, N. et al. (no date) 'Dropout: A simple way to prevent neural networks from overfitting', *The Journal of Machine Learning Research*. Available at: <http://citeseerx.ist.psu.edu/viewdoc/citations;jsessionid=62C00367CD502B74905266851BF65145?doi=10.1.1.696.4855> (Accessed: 29 August 2017).
- Suresh, H. et al. (no date) 'Clinical Intervention Prediction and Understanding using Deep Networks', pp. 1–16. Available at: <https://arxiv.org/pdf/1705.08498.pdf>.
- Wang, Z. et al. (2016) 'Representation Learning with Deconvolution for Multivariate Time Series Classification and Visualization'. Available at: <http://arxiv.org/abs/1610.07258> (Accessed: 31 August 2017).
- Yoon, J. et al. (2016) 'ForecastICU: A Prognostic Decision Support System for Timely Prediction of Intensive Care Unit Admission', *Proceedings of The 33rd International Conference on Machine Learning*, 48, pp. 1680–1689. Available at: <http://proceedings.mlr.press/v48/yoon16.html>.
- Zeiler, M. D. et al. (2010) 'Deconvolutional networks', in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2528–2535. doi: 10.1109/CVPR.2010.5539957.