Sum of previous inpatient serum creatinine measurements; a practical model for acute kidney injury in rehospitalized patients

Samuel J. Weisenthal1,2, Haofu Liao3, Philip Ng4, and Martin S. Zand1,2,5
1. University of Rochester School of Medicine and Dentistry, University of Rochester Medical Center
2. Rochester Center for Health Informatics
3. Department of Computer Science, University of Rochester
4. University of Rochester Information Systems Division
5. Department of Medicine, Division of Nephrology

Abstract

Acute Kidney Injury (AKI), the abrupt decline in kidney function due to temporary or permanent injury, is associated with increased mortality, morbidity, length of stay, and hospital cost. Sometimes, simple interventions such as medication review or hydration can prevent AKI. There is therefore interest in estimating risk of AKI at hospitalization. To gain insight into this task, we employ multilayer perceptron (MLP) and recurrent neural networks (RNNs) using serum creatinine (sCr) as a lone feature. We explore different feature input structures, including variable-length look-backs and a nested formulation for rehospitalized patients with previous sCr measurements. Experimental results show that the simplest model, MLP processing the sum of sCr, had best performance: AUROC 0.92 and AUPRC 0.70. Such a simple model could be easily integrated into an EHR. Preliminary results also suggest that inpatient data streams with missing outpatient measurements—common in the medical setting—might be best modeled with a tailored architecture.

1 Introduction

Acute Kidney Injury (AKI), the abrupt decline in kidney function due to temporary or permanent injury, is associated with increased mortality, morbidity, length of stay, and hospital cost [1]. There exist a variety of preventative strategies [2]—some types of AKI (e.g., from non-steroidal anti-inflammatory medication, radiocontrast chemotherapy, or aminoglycoside antibiotics) can be prevented outright by altering treatment or close monitoring. For this reason, there is particular interest in modeling AKI with electronic health record (EHR) data [3]. Prior hospitalizations generate enormous amounts of data, some of it inaccessible to clinicians via current interfaces; we hope to gain insight into the way this data might be leveraged to help predict and prevent AKI. Of particular interest as a predictor is serum creatinine (sCr). Creatinine is a protein that accumulates in the serum if kidney filtration is reduced, acting as a surrogate for the true measure of kidney function, glomerular filtration rate (GFR). In this study we process sCr with recurrent neural networks (RNNs) [4] and multilayer perceptrons (MLPs) using different input structures.

2 Related work and background

AKI prediction is an active area of research, with special emphasis on features from the electronic Health Record (EHR) data [5]. In particular, there is interest in construction of models that apply
to a broad patient population [3]. Many current models focus on AKI in the context of cardiac procedures, the critically ill, the elderly, liver and lung transplant patients, and rhabdomyolysis. Most use logistic regression, and although some use decision trees or ensemble methods. These models use features from the current hospitalization; in contrast, in line with [6], we use features from previous visits to estimate the probability of AKI in a rehospitalization given data from prior hospitalizations, focusing on the cohort of patients who are rehospitalized and also have previous sCr measurements. This particular cohort has high prior probability of AKI and therefore a predictive model could have real application; e.g., many rehospitalized patients present with conditions that may benefit from certain medications best administered when AKI risk is low. By considering only longitudinal sCr data, we also explore a much simpler, more interpretable model space than other studies. The EHR data flow is complex (Figure 1) and implementing models that depend on many features might be difficult, the models described here might more easily be implemented into an EHR, facilitating translation into the clinic.

Figure 1: EHR diagram. Blue lines are lab data flow, green billing and scheduling, and red medications. Dotted lines correspond to data files and solid to direct Health Level 7 integration.

RNNs are popular in medical research; they are used for phenotyping in [7] and heart failure detection and next visit prediction in [6,8]. In [6], the authors predicted AKI among other diagnosis codes, inspiring us to pursue it further with laboratory values in the inpatient setting. In [6], features are collected per unit of time, whereas here we only consider the order of measurements, but not their actual time stamps. Since we process sequences of measurements, there is related work in natural language modeling. As phrases can be represented by averaging words [9], hospitalizations might be represented by averaging measurements. Hierarchical RNN architectures have been employed [10,11] for context awareness and boundary inference.

3 Methods

We use an IRB-approved, de-identified adult dataset from an inpatient EHR system. The full dataset contains roughly six million laboratory records. We assigned a positive label if AKI was coded (according to International Statistical Classification of Diseases ICD 9th edition) or sCr trajectories met the most current diagnostic criteria [12]. The data for each patient consists of a variable-length sequence of hospitalizations, or hospitalizations, where each hospitalization is a variable-length sequence of measurements. We have no information on these patients outside of their hospitalizations. We restrict ourselves to a single feature, sCr to build a simple, interpretable model. We forward fill missing sCrs (since we processed sequences, there were very few of these, so we did not use a more sophisticated method; these were truly missing at random because a test was recorded, but no result value was present). We denote sCr measurement from hospitalization a at time t as \( m_{(a)}^{(t)} \). Each patient could therefore be represented as a list of \( A \) hospitalizations, each of which contained a list of \( \tau_{(a)} \) sCr measurements:

\[
[1 : [m_1^{(1)},\ldots,m_{(1)}^{(\tau_{(1)})}],\ldots, A : [m_1^{(1)},\ldots,m_{(A)}^{(\tau_{(A)})}]]
\]
There are $A$ labels, where each is an indicator of whether AKI occurred in the hospitalization directly following hospitalization $a$.

$$[l_1, \ldots, l_A]$$

### 3.1 Input feature structures

The basic formulations are as follows (further details are provided in the Appendix): each hospitalization-label pair can be considered independently (MARKOV); all prior measurements can be concatenated (CONCAT); we can consider all observations from the same patient as a single, nested sequence (NEST).

For processing by MLP, the measurements must be aggregated. We can choose MARKOV or CONCAT before aggregating and then employ MEAN, SUM, or MAX. Data processing was performed in Pandas \[13\].

### 3.2 Architectures

We do not review the standard RNN architecture in entirety, but we try to follow the notation in Goodfellow et al. \[14\], which provides more detail. For more detail on our methods, see the Appendix. We use cross-entropy loss and hyperbolic tangent activation in all experiments. We use a many-to-one RNN for MARKOV and CONCAT.

For the NEST input structure we require an RNN that processes nested, variable-length sequences where each inner sequence produces only a single output. We therefore modified the RNN architecture to process nested sequences by chaining together multiple instances of the many-to-one network. We suspected that information might be passed differently from one hospitalization to the next than it is from one measurement to the next, so we introduced a new “rehospitalization” parameter, $R$, that was only between hospitalizations. Since our data consists of hospitalization sequences that contain measurement sequences, we can index the new equations by $1 \leq a \leq A$ where each $a$ has $\tau(a)$ measurements. With $R$, we have the forward equations

$$z^{(t)}_{(a)} = \begin{cases} W_{m(a)}^{(t)} + R h^{(\tau(a-1))}_{(a-1)} + r & t = 1 \\ W_{m(a)}^{(t)} + U h^{(t-1)}_{(a)} + b & t \neq 1 \end{cases}$$

$$h^{(t)}_{(a)} = \tanh(z^{(t)}_{(a)})$$

$$y_{(a)} = V h^{(\tau(a))}_{(a)} + c$$

The network is now unrolled in time over measurements and over hospitalizations before backpropagation. All models were implemented using Numpy \[15\].

### 4 Experiments

The dataset contains 135,862 sCr measurements from 12,491 unique patients who generated 26,606 hospitalizations. AKI occurred in rehospitalization after 15.7% of the hospitalizations. Throughout the dataset, there were on average 2.1 ± 2.4 hospitalizations per patient and on average 5.1 ± 9.9 sCr measurements per hospitalization. Over 100 trials, we compare the three input structures MARKOV, CONCAT, and NEST where NEST has the inter-hospitalization variable $R$ (otherwise, it is identical to CONCAT). We also evaluate an MLP acting on MARKOV or CONCAT with aggregation function SUM, MEAN, or MAX. Each trial has identical (random) parameter initializations and the same shuffled training/validation dataset. Note that since the MLP has no parameter $U$ linking hidden states, we were not able to initialize it with the same parameters as the RNNs, but inter-MLP comparisons have identical parameter initializations as well. We explore different numbers of hidden units (HUs) (10, 50, and 100). Twenty percent of the full dataset was held out as test data and 20% of the training data was held out as validation data (the prevalence of AKI upon rehospitalization in the train and test sets were roughly equal at 16% and 15%, respectively). (20% is still held out.) For all

\[1\] We denote this one as MARKOV because it is memoryless past one hospitalization
splits, we ensure that no hospitalizations from the same patient are in the training and testing set. We trained each model for 20 epochs using AdaGrad [16] and then tested the model with the best validation set AUROC of the 100 trials on the test set. The distribution of area under the receiver operating characteristic (AUROC), area under the precision recall curve (AUPRC), and logistic loss (LL) for the validation set over the 100 trials with different numbers of hidden units are shown in the Appendix (Figure 2). We report untouched test set performance for the best models in Table 1. All metrics are computed at the hospitalization (not patient) level, and therefore patients with multiple hospitalizations are represented multiple times. Evaluation metrics were computed with functions from scikit-learn [17].

Table 1: Held-out test set performance

| # HU | Model | Input Struct | LL  | AUPRC | AUROC |
|------|-------|--------------|-----|-------|-------|
| 10   | RNN   | NEST         | 0.438376 | 0.460836 | 0.831527 |
|      |       | MARKOV       | 0.422072 | 0.614550 | 0.900993 |
|      |       | CONCAT       | 0.365777 | 0.469494 | 0.831687 |
|      | MLP   | MARKOV-MAX   | 0.373659 | 0.507971 | 0.843675 |
|      |       | CONCAT-MAX   | 0.381322 | 0.470231 | 0.832244 |
|      |       | MARKOV-MEAN  | 0.391130 | 0.416559 | 0.765329 |
|      |       | CONCAT-MEAN  | 0.392883 | 0.402276 | 0.758172 |
|      |       | MARKOV-SUM   | 0.281867 | 0.697588 | 0.919228 |
|      |       | CONCAT-SUM   | 0.394754 | 0.479620 | 0.844405 |
| 50   | RNN   | NEST         | 0.475782 | 0.430490 | 0.786414 |
|      |       | MARKOV       | 0.371971 | 0.564443 | 0.878528 |
|      |       | CONCAT       | 0.421936 | 0.213510 | 0.690209 |
|      | MLP   | MARKOV-MAX   | 0.375280 | 0.507971 | 0.843675 |
|      |       | CONCAT-MAX   | 0.383305 | 0.470231 | 0.832244 |
|      |       | MARKOV-MEAN  | 0.392830 | 0.416559 | 0.765314 |
|      |       | CONCAT-MEAN  | 0.394410 | 0.402274 | 0.758167 |
|      |       | MARKOV-SUM   | 0.281964 | 0.697587 | 0.919227 |
|      |       | CONCAT-SUM   | 0.374631 | 0.479623 | 0.844408 |
| 100  | RNN   | NEST         | 0.495673 | 0.471420 | 0.816330 |
|      |       | MARKOV       | 0.425833 | 0.596208 | 0.879722 |
|      |       | CONCAT       | 0.390785 | 0.211309 | 0.679677 |
|      | MLP   | MARKOV-MAX   | 0.373959 | 0.507971 | 0.843675 |
|      |       | CONCAT-MAX   | 0.380864 | 0.470231 | 0.832244 |
|      |       | MARKOV-MEAN  | 0.392619 | 0.416556 | 0.765317 |
|      |       | CONCAT-MEAN  | 0.393588 | 0.402281 | 0.758176 |
|      |       | MARKOV-SUM   | 0.282694 | 0.697584 | 0.919224 |
|      |       | CONCAT-SUM   | 0.387263 | 0.479623 | 0.844405 |

HU = hidden units; LL = log loss; AUPRC = area under PR curve; AUROC = area under ROC curve

5 Conclusion

Using only sCr, RNNs and MLP predict AKI in rehospitalizations with high AUROC and AUPRC. The best performing model was very simple. MARKOV, where we treated each hospitalization as an independent sequence, was the best performing input structure for the RNN; MARKOV-SUM, where we treated each hospitalization as an independent sequence and aggregated it using sum, gave best results for the MLP and in general. Although this is still preliminary work and requires validation at different institutions, this is an exceedingly simple model and might be easily integrated into an EHR (since performance was insensitive to HU, a lower-capacity model like logistic regression could be appropriate). Further, it should be noted that this model was actually a baseline and therefore its discovery was happenstance—it was not the original intent of the authors to evaluate it; a new study intended to externally validate the model would help ensure that the finding was not unique to this particular dataset. Such a model is enticing however because the sum of sCr takes into account both sCr values—reflecting renal function, frequency of the orders— reflecting physician concern, and length of the hospitalization (more work should be done to disentangle these factors, but capturing them all in a single input allows us to only estimate one parameter).

Our study also sheds light on how to model inpatient data when outpatient measurements are unknown. This is a common occurrence in medical datasets, many of which are generated by a single hospital lacking complete outpatient data. For 10 HU, inserting a different parameter R between hospitalizations did not seem to affect results. For greater than 10 HU, inserting R appeared to improve AUROC and AUPRC considerably but worsen LL. LL might have decreased because AUROC, not LL, was optimized for in model selection and the selected model was not required to be properly calibrated, but further experiments are needed to more fully explore this phenomenon.
These preliminary results however might suggest that a similar architecture or hierarchical RNN might be best for data streams with missing outpatient measurements—information flows differently between measurements than between hospitalizations.

We stress our general finding of the benefits of investigating different permutations of input data structure (varying time window and aggregation function) and models for medical time-series prediction tasks. The simple models yielded from this process might be very interpretable (simply summing all previous sCr measurements is easy to explain and could even be performed manually) and easy to implement into an EHR, which could allow quick adoption and facilitate the transition to automated “data-driven” healthcare practice, making way for more sophisticated techniques in the future. Future directions include incorporating time stamps because more recent measurements might be most important and employing more sophisticated RNN architectures.

Acknowledgments

The project described in this publication was supported by the University of Rochester CTSA award number TL1 TR 002000 from the National Center for Advancing Translational Sciences of the National Institutes of Health. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors acknowledge Robert White for obtaining the data and the anonymous reviewers for help in improving the manuscript.

Appendix

Input structures

- Each hospitalization-label pair can be considered independently (MARKOV).
  \[\left[\left[m^{(1)}_{(1)} \ldots, m^{(\tau_{(1)})}_{(1)}\right], \left[l_{1}\right]\right], \ldots, \left[\left[m^{(1)}_{(A)} \ldots, m^{(\tau_{(A)})}_{(A)}\right], \left[l_{A}\right]\right]\]

  This corresponds to the time window approach with lag 1 visit in [18]. This kind of Markovian approach ignores measurements from all but the most recent hospitalization, and also does not take into account that hospitalizations are from the same patient. Each sequence of measurements, however, is relatively short, alleviating concerns about vanishing or exploding gradients [19, 20].

- All prior measurements can be concatenated (CONCAT).
  \[\left[\left[m^{(1)}_{(1)} \ldots, m^{(\tau_{(1)})}_{(1)}\right], \left[l_{1}\right]\right], \ldots, \left[\left[m^{(1)}_{(1)} \ldots, m^{(\tau_{(1)})}_{(1)} \ldots, m^{(1)}_{(A)} \ldots, m^{(\tau_{(A)})}_{(A)}\right], \left[l_{A}\right]\right]\]

  Unlike MARKOV, CONCAT has memory for previous hospitalization measurements, but it still ignores that these hospitalizations are from the same patient.

- We can consider all observations from the same patient as a single, nested sequence (NEST).
  \[\left[\left[m^{(1)}_{(1)} \ldots, m^{(\tau_{(1)})}_{(1)}\right], \ldots, \left[\left[m^{(1)}_{(1)} \ldots, m^{(\tau_{(1)})}_{(1)} \ldots, m^{(1)}_{(A)} \ldots, m^{(\tau_{(A)})}_{(A)}\right], \left[l_{1}, \ldots, l_{A}\right]\right]\]

  NEST should theoretically be preferable to MARKOV and CONCAT as all measurement information for each hospitalization is retained and also we take into account that the hospitalizations are from the same patient. An RNN processing NEST requires nBPTT for training. NEST produces a single sequence of the same length as the longest sequence in CONCAT.

For processing by MLP, the measurements must be aggregated. We can choose MARKOV or CONCAT before aggregating and then employ MEAN, SUM, or MAX.

For example, to aggregate a MARKOV hospitalization using SUM, we just convert:

\[\left[m^{(1)}_{(j)} \ldots, m^{(\tau_{(1)})}_{(j)}\right] \rightarrow \sum_{i=1}^{\tau_{(1)}} m^{(i)}_{(j)}\]

Benefits of the aggregation approach are that it is parsimonious and SUM elegantly represents the number of measurements (weighted by magnitude)—this could be highly predictive since the frequency of medical testing often reflects a clinician’s anxiety over a patient’s status, which might be a strong indicator of deterioration, but the value of the test is also relevant.
RNNs

Again we follow the notation in Goodfellow et al. [14]. We only process a single scalar sCr, but provide the matrix equations in case of multi-dimensional input. Each sCr measurement is denoted \( m \). We use a many-to-one RNN for MARKOV and CONCAT. Given a loss \( L \) (we use cross entropy), an initial state \( h^{(0)} \), and \( T \) timesteps each with a sCr measurement \( m^{(t)} \), the forward equations with hyperbolic tangent activation are

\[
\begin{align*}
    z^{(t)} &= W m^{(t)} + U h^{(t-1)} + b \\
    h^{(t)} &= \tanh(z^{(t)}) \\
    y^{(t)} &= V h^{(t)} + c
\end{align*}
\]

Relative to a many-to-many RNN, the gradient of the hidden state becomes

\[
\nabla_{h^{(t)}} L = \left\{ \begin{array}{ll}
    (\frac{\partial y}{\partial h^{(t)}})^T \nabla_y L &= V^T \nabla_y L & t = \tau \\
    (\frac{\partial h^{(t+1)}}{\partial h^{(t)}})^T \nabla_{h^{(t+1)}} L &= U^T J \nabla_{h^{(t+1)}} L & t \neq \tau
    \end{array} \right.
\]

Where \( J \) is the Jacobian of the hyperbolic tangent.

For the NEST input structure, we require an RNN that processes nested, variable-length sequences where each inner sequence produces only a single output. We therefore construct an RNN for nested sequences by chaining together multiple instances of the many-to-one network (similar to a hierarchical RNN [11]). We suspect that information might be passed differently from one hospitalization to the next than one measurement to the next, so we introduce a new "rehospitalization" parameter, \( R \), that is only between hospitalizations. Since our data consists of hospitalization sequences that contain measurement sequences, we can index the new equations by \( 1 \leq a \leq A \) where each \( a \) has \( \tau_{(a)} \) measurements. As mentioned in the body of the paper, with \( R \), we have the forward equations

\[
\begin{align*}
    z_{(a)}^{(t)} &= \left\{ \begin{array}{ll}
    W m_{(a)}^{(t)} + R h_{(a-1)}^{(\tau_{(a-1)})} + r & t = 1 \\
    W m_{(a)}^{(t)} + U h_{(a)}^{(t-1)} + b & t \neq 1
    \end{array} \right.
\]

\[
    h_{(a)}^{(t)} = \tanh(z_{(a)}^{(t)}) \\
    y_{(a)} = V h_{(a)}^{(\tau_{(a)})} + c
\]

The loss is now summed over hospitalizations from the same patient

\[
L = \sum_a L_{(a)}
\]

The network is now unrolled in time over measurements and over hospitalizations before back-propagation [21] (automatic differentiation [22] could also be used here). Since we use at each hospitalization an RNN with a single output, the hidden state gradient is

\[
\nabla_{h_{(a)}^{(t)}} L = \left\{ \begin{array}{ll}
    (\frac{\partial h_{(a)}^{(t+1)}}{\partial h_{(a)}^{(t)}})^T \nabla_{h_{(a)}^{(t+1)}} L &= U^T J \nabla_{h_{(a+1)}^{(t+1)}} L & t \neq \tau_{(a)} \\
    (\frac{\partial y_{(a)}}{\partial h_{(a)}^{(t)}})^T \nabla_{y_{(a)}} L + (\frac{\partial h_{(a+1)}^{(1)}}{\partial h_{(a)}^{(t)}})^T \nabla_{h_{(a+1)}^{(1)}} L &= V^T \nabla_{y_{(a)}} L + R^T J \nabla_{h_{(a+1)}^{(1)}} L & t = \tau_{(a)}, a \neq A \\
    \frac{\partial y_{(a)}}{\partial h_{(a)}^{(t)}})^T \nabla_{y_{(a)}} L &= V^T \nabla_{y_{(a)}} L & t = \tau_{(a)}, a = A
    \end{array} \right.
\]
The parameter gradients, collected now over time and hospitalizations, are accessible through the node gradients

$$\nabla_c L = \sum_a \left( \frac{\partial y^{(\tau(a))}_{(a)}}{\partial c^{(\tau(a))}_{(a)}} \right)^T \nabla_{y^{(\tau(a))}_{(a)}} L = \sum_a \nabla_{y^{(\tau(a))}_{(a)}} L$$

$$\nabla_V L = \sum_a \nabla_{y^{(\tau(a))}_{(a)}} L \left( \frac{\partial y^{(\tau(a))}_{(a)}}{\partial V^{(\tau(a))}_{(a)}} \right) = \sum_a \nabla_{y^{(\tau(a))}_{(a)}} L h^{(\tau(a))}_{(a)} T$$

$$\nabla_b L = \sum_a \sum_{t>1} \left( \frac{\partial h^{(t)}_{(a)}}{\partial b^{(t)}_{(a)}} \right)^T \nabla_{h^{(t)}_{(a)}} L = \sum_a \sum_{t>1} J \nabla_{h^{(t)}_{(a)}} L$$

$$\nabla_W L = \sum_a \sum_{t} \nabla_{h^{(t)}_{(a)}} L \left( \frac{\partial h^{(t)}_{(a)}}{\partial W^{(t)}_{(a)}} \right) = \sum_a \sum_{t} J \nabla_{h^{(t)}_{(a)}} L m^{(t)} T$$

$$\nabla_U L = \sum_a \sum_{t>1} \nabla_{h^{(t)}_{(a)}} L \left( \frac{\partial h^{(t)}_{(a)}}{\partial U^{(t)}_{(a)}} \right) = \sum_a \sum_{t>1} J \nabla_{h^{(t)}_{(a)}} L h^{(t-1)}_{(a)} T$$

$$\nabla_R L = \sum_a \nabla_{h^{(1)}_{(a)}} L \left( \frac{\partial h^{(1)}_{(a)}}{\partial R^{(1)}_{(a)}} \right) = \sum_a J \nabla_{h^{(1)}_{(a)}} L h^{(\tau(a-1))}_{(a-1)} T$$

$$\nabla_r L = \sum_a \nabla_{h^{(1)}_{(a)}} L \left( \frac{\partial h^{(1)}_{(a)}}{\partial r^{(1)}_{(a)}} \right) = \sum_a J \nabla_{h^{(1)}_{(a)}} L$$
Validation Set Distributions

![Image of distributions for 10, 50, and 100 hidden unit MLP and RNN processing different inputs structures over 100 trials. HU = hidden units; LL = log loss; AUPRC = area under PR curve; AUROC = area under ROC curve.]

Figure 2: Distributions of validation set AUROC, AUPRC, and LL for 10, 50, and 100 hidden unit MLP and RNN processing different inputs structures over 100 trials. HU = hidden units; LL = log loss; AUPRC = area under PR curve; AUROC = area under ROC curve

References

[1] Chertow GM, Burdick E, Honour M, Bonventre JV, Bates DW. Acute kidney injury, mortality, length of stay, and costs in hospitalized patients [Journal Article]. J Am Soc Nephrol. 2005;16(11):3365–70. Available from: [Link](https://www.ncbi.nlm.nih.gov/pubmed/16177006).

[2] Lameire N, van Biesen W, Hoste E, Vanholder R. The prevention of acute kidney injury an in-depth narrative review: Part 2: Drugs in the prevention of acute kidney injury [Journal Article]. NDT Plus. 2009;2(1):1–10. Available from: [Link](https://www.ncbi.nlm.nih.gov/pubmed/25949275).

[3] Sutherland SM, Chawla LS, Kane-Gill SL, Hsu RK, Kramer AA, Goldstein SL, et al. Utilizing electronic health records to predict acute kidney injury risk and outcomes: workgroup statements from the 15(th) ADQI Consensus Conference [Journal Article]. Can J Kidney Health Dis. 2016;3:11. Available from: [Link](https://www.ncbi.nlm.nih.gov/pubmed/26925247).
[4] Rumelhart DE, Smolensky P, McClelland JL, Hinton G. Sequential thought processes in PDP models. Parallel distributed processing: explorations in the microstructures of cognition. 1986;2:3–57.

[5] Koyner JL, Adhikari R, Edelson DP, Churpek MM. Development of a Multicenter Ward-Based AKI Prediction Model [Journal Article]. Clin J Am Soc Nephrol. 2016;11(11):1935–1943. Available from: https://www.ncbi.nlm.nih.gov/pubmed/27633727

[6] Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor ai: Predicting clinical events via recurrent neural networks. In: Machine Learning for Healthcare Conference; 2016. p. 301–318.

[7] Lipton ZC, Kale DC, Elkan C, Wetzell R. Learning to diagnose with LSTM recurrent neural networks. arXiv preprint arXiv:151103677. 2015;

[8] Choi E, Schuetz A, Stewart WF, Sun J. Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association. 2016;24(2):361–370.

[9] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013;

[10] Sordoni A, Bengio Y, Vahabi H, Lioma C, Grue Simonsen J, Nie JY. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM; 2015. p. 553–562.

[11] Chung J, Ahn S, Bengio Y. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:160901704. 2016;

[12] Ad-hoc working group of E, Fliser D, Laville M, Covic A, Fouque D, Vanholder R, et al. A European Renal Best Practice (ERBP) position statement on the Kidney Disease Improving Global Outcomes (KDIGO) clinical practice guidelines on acute kidney injury: part 1: definitions, conservative management and contrast-induced nephropathy [Journal Article]. Nephrol Dial Transplant. 2012;27(12):4263–72. Available from: https://www.ncbi.nlm.nih.gov/pubmed/23045432

[13] McKinney W, et al. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. vol. 445. van der Voort S, Millman J; 2010. p. 51–56.

[14] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

[15] Walt Svd, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering. 2011;13(2):22–30.

[16] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research. 2011;12(Jul):2121–2159.

[17] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825–2830.

[18] Choi E, Bahadori MT, Sun J. Doctor ai: Predicting clinical events via recurrent neural networks. arXiv preprint arXiv:151105942. 2015;

[19] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks. 1994;5(2):157–166.

[20] Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press; 2001.

[21] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Cognitive modeling. 1988;3(3):1.

[22] Rall LB. Automatic differentiation: Techniques and applications. 1981;