UNIVERSITY OF CALIFORNIA

Los Angeles

Deep Risk: Timely Risk Scoring by a Recurrent Ensemble of Recurrent Neural Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical & Computer Engineering

by

Anton Nemchenko

2018

ABSTRACT OF THE THESIS

Deep Risk: Timely Risk Scoring by a Recurrent
Ensemble of Recurrent Neural Networks

by

Anton Nemchenko
Master of Science in Electrical & Computer Engineering
University of California, Los Angeles, 2018
Professor Mihaela Van Der Schaar, Chair

Timely prediction of clinical adverse events is a ubiquitous and important problem. We present here a timely risk scoring algorithm (Deep Risk) based on a novel Deep Learning architecture that solves the following key challenges: 1) the statistical properties of the physiological time-series data streams are not constant over time; 2) timely prediction is of the essence; 3) different patients exhibit different physiological trajectories; 4) the data is unbalanced (adverse events are uncommon). Deep Risk employs a Gated Recurrent Unit (GRU)-based Recurrent Neural Network (RNN) to aggregate the predictions of a family of GRU-based RNN's which operate on time windows of varying lengths. We show that shorter windows cope better with the non-stationary data but longer windows are capable of issuing more timely predictions. Using both shorter and longer windows enables Deep Risk to do both. Each "lower level" RNN uses the information in its time window to make a prediction; the "higher level" RNN uses the information in the longest of these time intervals to aggregate the predictions of the "lower level" RNN's and issue a final prediction. We perform simulations based on real-world medical data sets and show that Deep Risk achieves large and significant performance improvement over other methods, including clinical risk scores and state-of-the-art machine learning algorithms.

The thesis of Anton Nemchenko is approved.

Vwani P. Roychowdhury

Gregory J. Pottie

Mihaela Van Der Schaar, Committee Chair

University of California, Los Angeles

2018

To my family . . .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Every year in the U.S., more than 200,000 hospitalized patients experience cardio-pulmonary arrest, 75% of those patients die and 50% of those who die could have been saved by timely interventions, especially admission to an Intensive Care Unit (ICU). This represents more than 75,000 unnecessary deaths in hospital every year. Evidently, the warning systems in operation are not up to the task, which explains why enormous efforts have been expended to create better warning systems. Unfortunately, none of these efforts have been as successful as might have been hoped. In terms of True Positive Rate (TPR)/Sensitivity vs Positive Predictive Value (PPV)/Precision, the most commonly used clinical risk scoring systems (MEWS [SKR01] and APACHE [KDW85]) achieve Area Under the Sensitivity/Precision Curve (AUSPC) of only 0.172 and the state-of-the-art machine learning scoring system [AYH17] achieves AUSPC of only 0.361. This paper reports a new risk scoring system, Deep Risk, based on novel Deep Learning architecture, that achieves AUSPC of 0.460.

In recent years, an enormous amount of effort has been expended to develop machine learning methods for the prediction of various adverse events (in the hospital): admission to ICU [AYH17], septic shock [HHP15], etc. The approaches taken have included Neural Networks [KCB15], [CKL15], Gaussian Processes [CCP12], Hidden Markov Models [AYH17] [HV16], hypothesis testing [YAH16]. Although these methods have generally improved on state-of-the-art clinical methods, there is much room for improvement.

The lack of success of work to date comes from the many very difficult challenges the problem poses. The first is that the statistical properties (for instance, the mean and variance) of the physiological time-series data streams are not constant over time; they are *non-stationary* [AYH17]. This is important because a single patient may exhibit different patterns over time. The second is that timely prediction is of the essence: intervention may be required *now*; delay of even a few hours may prove irreversible or even fatal. It follows that to be useful, a warning must be issued well in advance of the time it becomes obvious (to the clinicians) that intervention or admission to ICU is required. The third is that different patients may exhibit different patterns over time. (Identifying groups of patients for which disease patterns are similar – i.e., phenotyping – has proved to be very important in other contexts, and numerous papers have used machine learning techniques to attempt to use phenotyping for predicting adverse events [CKL15, AYH16, SS15]. However these papers have not succeeded in learning temporal phenotypes in settings in which the disease pattern (trajectory) is non-stationary and in which both static and dynamic data from many sources needs to be integrated.) The fourth is that the data is very unbalanced: only a small fraction of patients experience adverse events. (For instance: only 5% of hospitalized patients will require transfer to ICU.).

We address these challenges by using a novel Deep Learning methodology that employs a Gated Recurrent Unit (GRU)-based Recurrent Neural Network (RNN) to aggregate the predictions of a family of GRU-based RNN Denoising Autoencoders (DAE). Each of the "lower level" RNN's uses the information in a time window of a specific length (e.g., 4 hours before the present time, 8 hours before the present time, etc.) to make a prediction; the "higher level" RNN uses the information in the longest of these time intervals to aggregate the predictions of the "lower level" RNN's and issue a single (final) prediction. Each of the "lower level" RNN's learns *what* information is important in its time window; the "higher level" RNN learns *when* the information is important; this addresses the first challenge. We address the second challenge by using GRU's with different time windows to issue both current predictions and future predictions (e.g. the patient will need to be sent to ICU 8 hours from now). We address the third challenge by using the DAE's to produce latent representations/phenotypes. However, rather than using the DAE to learn general *unsupervised* representations/phenotypes and then using other machinery to issue a prediction for each phenotype, we integrate the DAE with a GRU-based classification layer so that we learn a task specific, mixed *supervised* and *unsupervised* representation. (Our DAE's add noise to the input time series and then reconstruct the original input from the noisy version. A novel aspect of our construction is that we reconstruct *both* the input at the current time and the input at the next (future) time and penalize for errors in each of these reconstructions and in the correlation between them. This regularization allows us to learn more predictive representations.) We address the imbalance of the data set by bootstrapping (re-sampling the data) to enable the RNN to learn more accurately for the small fraction of patients who experience adverse events.

We demonstrate these results using the same data set used by [AYH17] which has a heterogeneous cohort of patients hospitalized in a large medical center. In comparison with clinical risk scoring systems and previous machine learning risk scoring systems Deep Risk achieves very large improvements in AUSPC (TPR vs PPV) and significant improvements in AUROC (TPR vs FPR). With respect to correct prediction at specific times, Deep Risk achieves truly spectacular improvements. For instance, in terms of predictions 8 hours before actual clinical decisions to admit to ICU, at a fixed TPR of 50%, the best clinical risk scores achieve PPV below 0.15, the best previous machine learning score achieves PPV of 0.26 and Deep Risk achieves PPV of 0.70. Taken together, these performance numbers show that Deep Risk is able to provide recommendations that are better, timelier and have greater reliability than those provided by clinical risk scoring systems and previous machine learning risk scoring systems.

CHAPTER 2

Related Work

The paper closest to ours in terms of problem, intent and data is [AYH17]. However, [AYH17] takes a very different approach based on multi-task Gaussian processes. In the Results Section, we directly compare the performance of Deep Risk to that of [AYH17] and show that Deep Risk achieves much superior performance. This is due, among other things, to its ability to effectively integrate static and dynamic features and implicitly learn phenotypes (thereby avoiding model-selection difficulties) in order to issue predictions.

[KCB15] consider a setting similar to ours, with time series and static features. This paper first learns phenotypes using a multilayer DAE and then uses the phenotypes to make predictions using a sigmoid layer. Because an RNN is not used, the temporal relationships between samples are not exploited. The thrust of [CKL15] is to identify phenotypes using a feedforward network with prior-based regularization. The temporal relationships are captured by expanding the network at each time step and initializing the new weights using the weights constructed at the previous time step, either by similarity or through Gaussian sampling. The focus of the paper is on the identification of disease phenotypes and not on prediction of adverse events. [RMS16] predicts the onset of a disease months in the future (rather than an adverse event). The neural network structure used for prediction is very different from ours: they use two convolutional architectures and a standard Long Short-Term Memory unit-based RNN. They deal with the unbalancedness of the data by using a weighted cost function that assigns higher weights to rare diseases rather than using bootstrap resampling as we do. [CBS16] predicts diagnosis, medication and the time to follow-up, but does not treat the non-stationarity of the data. It uses a standard RNN architecture. [PTP16] models long-term disease trajectories and issues long-term risk predictions from infrequently gathered data; this poses a very different set of challenges. Non-stationarity, phenotyping and timely prediction of adverse events are not in the scope of this work.

CHAPTER 3

Goal

We begin by formalizing our approach to the problem of timely risk prediction and our goal.

We have a dataset with N patients. For each patient n we have a vector process $\{X_n(t)\}$ of *features*, including *static features* such as gender, age, height and weight, ICD9 codes, which are recorded when a patient is admitted and do not change during the hospital stay, and *dynamic features*, such as vital signs and lab tests, which do change during the hospital stay. The dynamic features are irregularly sampled and the time duration is different for different patients (who have different lengths of stay in the hospital). We write $x(t)$ for the realization of this process for a particular patient and $x_i(t)$ for the value of the data stream i at time t for this patient. For each patient, there is an eventual *outcome*: either an adverse event (e.g. admission to the ICU) or discharge from the hospital. We write $Y = 0$ for discharge and $Y = 1$ for the adverse event.

As with other algorithms, the goal of Deep Risk is to use the available data to issue a *risk score*. As usual, to evaluate the risk score we set a threshold τ and treat a risk score above τ as the prediction of an adverse event. Among those predictions, the true positives are those for which the adverse event actually occurred and the false positives are those for which the event did not occur; the ratio of the number of true positives to the number of adverse events is the *sensitivity* or *true positive rate* (TPR) and the ratio of the number of false positives to the number of adverse events is the *false positive rate* (FPR). The ratio of the number of true positives to the number of predicted adverse events is the *precision* or *positive predicted value* (PPV). Hence every threshold τ yields a pair of numbers $(\text{TPR}(\tau), \text{PPV}(\tau))$; varying τ yields the Sensitivity vs Precision curve.

Our goal in this paper is to issue accurate and timely predictions of the eventual outcome.

We measure accuracy using TPR vs PPV; it is argued that this measure is more informative than other measures in the evaluation of binary classifiers on unbalanced data sets such as ours [SR15]. We do this in different ways. As in [AYH17], we set a threshold, compute TPR and PPV at the first time the computed risk exceeds that threshold and then plot TPR and PPV for different values of the threshold. This is an important measure but it is not the only important measure. We use TPR vs PPV to measure timeliness both for predictions for different horizons into the future and for predictions within the last 24 hours. We also measure timeliness in terms of prediction in advance of actual admission. Finally, we also measure the performance Deep Risk using the conventional AUROC curve (TPR vs FPR).

CHAPTER 4

Deep Risk: Algorithm

The Deep Risk algorithm operates in two modes. In the offline mode, the risk scoring model is learned from the training data; in the online mode, risk scores are computed as needed for a hospitalized patient. Note that risk scores for a given patient may be computed at many times, as new information about the patient is acquired. We first describe the offline mode and then the online mode.

4.1 Offline Mode

Figure 4.1 displays the architecture of Deep Risk. For each time window there is a GRU-based DAE that learns a latent representation and a classifier that turns this representation into a recommendation/risk score. Then there is another GRU-based RNN that combines the recommendations from the classifier's for the various windows into a single recommendation/risk score. Overall, the architecture is a RNN of RNN's.

Since our algorithm is based on GRU's, we first describe these briefly and then particularize to our construction. Denote the logistic sigmoid function by $\sigma(\cdot)$ and a general activation function by $act(\cdot)$. Corresponding to each data stream i , there is an output function $h_i(t)$ (memory state) that follows the recursive equation:

$$h_i(t) = (1 - z_i(t))h_i(t-1) + z_i(t)act([Wx(t) + U(r(t) \odot h(t-1))]_i)$$

Thus $h_i(t)$ is a linear combination of the output $h_i(t-1)$ at the previous time and an update at the current time. The update in turn is an activation of a weighted combination of the current input $x(t)$ and the product of the previous memory state with the reset gate $r(t)$. The amount of update is regulated by the update gate $z_i(t)$. The reset and the update gates follow the equations:

$$
\begin{aligned}
r_i(t) &= \sigma([W^r x(t) + U^r h(t-1)]_i) \\
z_i(t) &= \sigma([W^z x(t) + U^z h(t-1)]_i)
\end{aligned}
$$

These equations involve the learnable parameters: (W, U), (W^z, U^z), (W^r, U^r), which are optimized using back-propagation via stochastic gradient descent.

We use different activation functions for different types of layers. For regression layers, we use the linear activation $act(x) = x$; for classification layers, we use the logistic sigmoid function $act(x) = \sigma(x) = \frac{1}{1+e^{-x}}$; and for the hidden layers we use the rectified linear unit (ReLU) $act(x) = max(0, x)$.

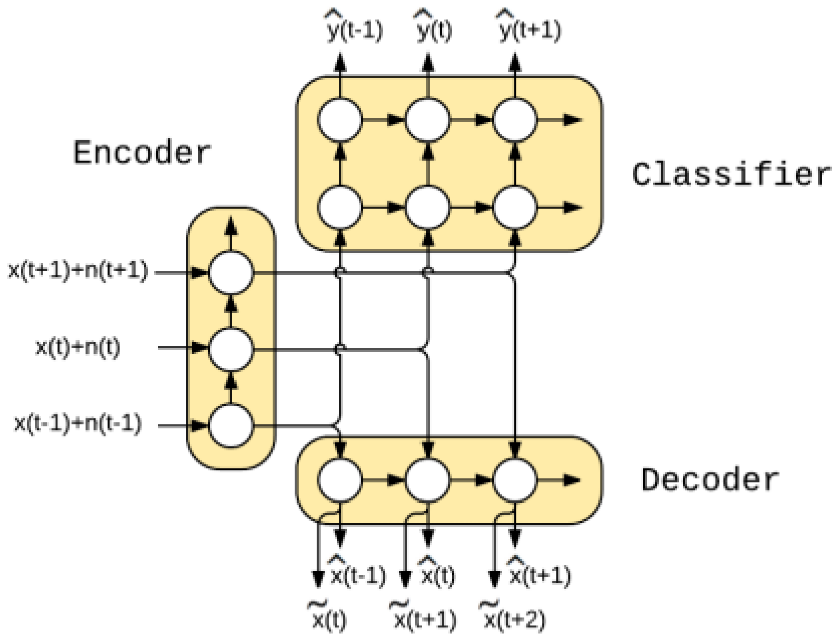

Figure 4.1: Deep Risk: architecture.

As we mentioned earlier, an essential component in our algorithm is a DAE. This component works by adding noise (with zero mean) to the input time series and then reconstructing the original input from the noisy version. A DAE has two parts: an encoder and a decoder. Our encoder is a single GRU layer with a ReLU activation and width that is 3 times the number of dynamic features; the decoder is a single GRU layer with a linear activation and width that is 2 times the number of dynamic features. The encoder learns the latent representation which is then decoded to reconstruct the original input data and to predict future data. (We will elaborate shortly).

The typical way to employ a DAE is to train it in an unsupervised manner and then add a classification layer on top of the DAE [KCB15]. When employed in this way, the DAE learns a general latent representation of the data that is not optimized for a specific task. However, we would like our DAE to learn a *specific latent representation* of the data that is *optimized for our specific task (prediction)*. To do this we train the DAE jointly with the classification layer and directly optimize the latent representation. In our case, the classification layer is a two layer, GRU-based RNN. The first layer uses a ReLU activation and is fed with the encoded time series (containing both the static and dynamic features), its width is 4 times the number of dynamic features. The second layer is a single, sigmoid activated, GRU hidden unit that outputs the risk score. (Recall that we use a separate DAE-GRU and issue a separate risk score for each time window; we discuss aggregation below.) Since we are feeding the DAE with noisy data, we would like to make sure that the representation the encoder learns is robust to noise and can be effectively used in the classification layer. We achieve this goal by temporal regularization: We reconstruct the *current* input and predict the *next* input, measure the errors in predictions of outcome and reconstructions of the data, and compute a loss function that takes all of these errors into account. To be precise, let $\hat{x}(t)$ be the reconstructed input for time t that our algorithm is producing at time t when fed with $\{x(s)+n(s) \mid s \leq t,\ n(s) \sim N(0,\sigma^2)\}$, let $\tilde{x}(t+1)$ be the prediction at time t of the input at time $t+1$ and let $\hat{y}(t)$ be the predicted risk score at time t. The time regularized loss function is:

$$L(t) = -y(t)log\left[\hat{y}(t)\right] - (1-y(t))\,log\left[(1-\hat{y}(t))\right]$$

$$+\alpha\left\|\hat{x}(t) - x(t)\right\|_2^2 + \beta\left\|\tilde{x}(t+1) - x(t+1)\right\|_2^2$$

$$+\gamma\left\|\left[\tilde{x}(t+1) - x(t+1)\right] + \left[\hat{x}(t) - x(t)\right]\right\|_2^2$$

where α, β, γ are tradeoff parameters. Note that the first line (i.e. the logarithmic terms) is the familiar cross entropy. The remaining terms are penalties: α multiplies (the square of the norm of) the error vector of the reconstruction at time t; β multiplies (the square of the norm of) the error vector of the prediction at time t of the value at time $t+1$; γ multiplies (the square of the norm of) the sum of these error vectors. Notice that the size of the γ term depends on the magnitude of the two error vectors *and* on the angle between them: if they point in the same direction, the errors simply add; otherwise there is some cancellation. Hence this term represents a proxy for the correlation between the measurement errors at time t and at time $t+1$. We optimize the parameters by cross validation.

This regularized loss function plays an important role in the performance of Deep Risk because appropriate choices of the coefficients α, β, γ allow the loss function to control the accuracy and consistency of the representation as well as the accuracy of the prediction. This is important because more accurate and consistent representations ultimately lead to better predictions. The effect of regularization will be discussed in the Results Section.

As we will discuss in Subsection 4.3, to balance the non-stationarity of the data against the need for timely predictions, we will form predictions using classifiers with different window lengths and then aggregate the predictions of these classifiers using an *ensembler*. The ensembler is a 2-layer GRU-based RNN. The first layer is ReLU-activated and has width that is 3 times the number of features; the second is a softmax layer of width equal to the number of classifiers. The ensembler computes weights for the individual classifiers and aggregates them according to these weights to issue a final risk score.

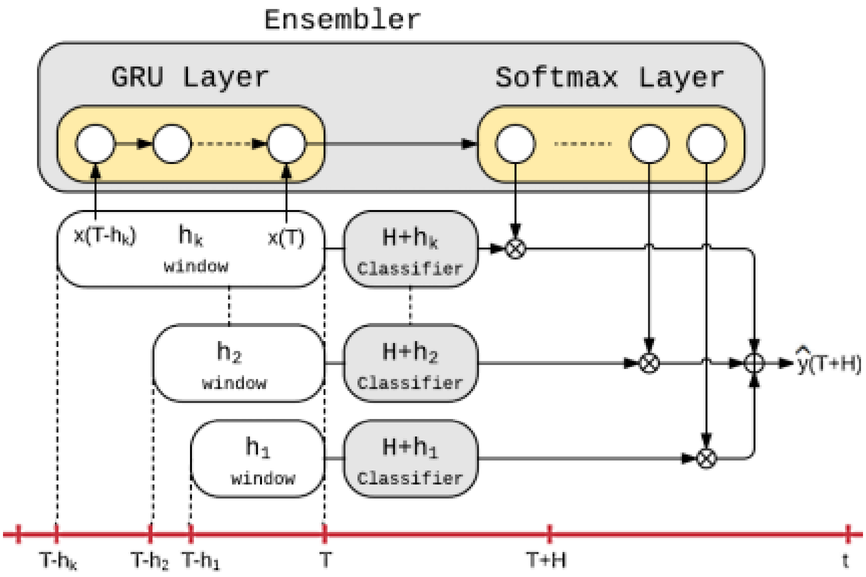

Figure 4.2: Deep Risk: run-time operation

4.2 Run-time

At run-time, we issue predictions for a *specified* number of hours ahead; the specified number of hours will be chosen by the clinician. This is an especially challenging but useful task because it incorporates *urgency*. Notice that letting H tend to infinity yields a prediction of the ultimate outcome for the patient; this is the predictive task considered by much of the literature, including [AYH16]. Evidently, issuing predictions for every horizon H is more difficult than simply issuing a prediction in the limit. To issue at the present time T a prediction through time $T + H$ (H hours ahead) we choose $h_1 < h_2 \ldots < h_K$; the k-th classifier uses the time window from time $T - h_k$ to $T + H$, so the k-th classifier has a window of length $h_k + H$. The information provided to the k-th classifier is simply the information available at time T; i.e. the information from time $T - h_k$ to the present time T. We then have this classifier issue a prediction. (Note that this treats the time from T to $T + H$ as if there were no measurements so non-causal data is not used.) The ensembler then uses the data from time $T - h_K$ (the longest) to the present time T and the predictions of all the classifiers to issue the prediction for H hours ahead. This is illustrated in Figure 4.2.

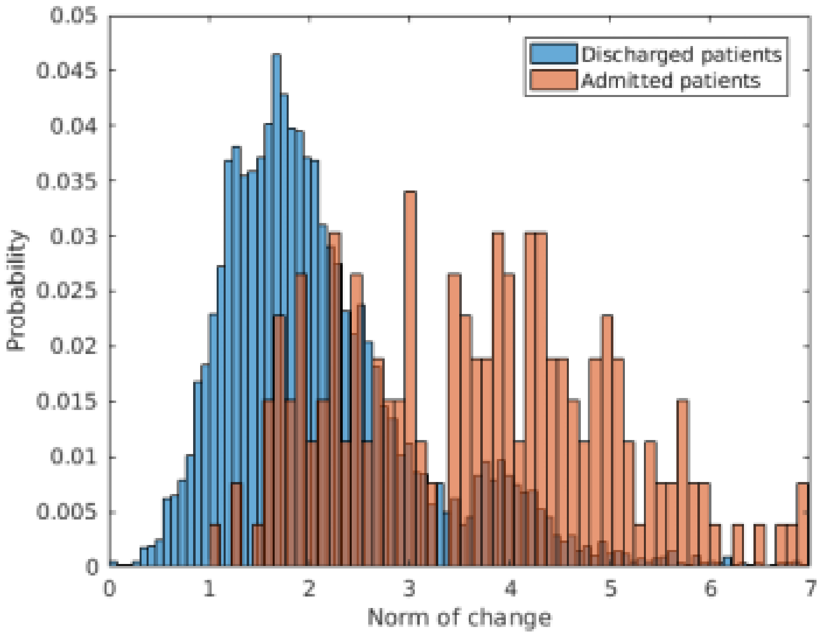

Figure 4.3: (Norm of) Change in patient features in the last 24 hours before decision

4.3 Window Lengths and Predictions

What window length should we use? If the data generating process were stationary, we would always prefer a longer window because that would reduce the likelihood of seeing an unrepresentative sample. But the data generating process *is not stationary*: the measurements for patients who will go the the ICU are deteriorating while the measurements for patients who will be discharged are stable, or perhaps improving. This can be seen in Figure 4.3 which provides histograms of the (norm of) the change in the feature vector of patients in the last 24 hours before discharge/ICU. (For each bar, the vertical axis shows the probability that a patient would experience a features change of that magnitude, conditional on that patient being of the type identified.)

Thus, newer information is more important. If we asked each classifier to issue a prediction only at the final (discharge/ICU) time, the classifier would learn to upgrade newer information and downgrade older information. However, because we ask each classifier to issue a prediction at *every time*, the process of downgrading old information is slow. Taken together, these forces suggest that shorter windows might provide more accurate predictions.

As Table 4.1 shows, this is exactly what we see: shorter windows do indeed provide more accurate predictions (in terms of AUSPC for TPR vs PPC).

Table 4.1: Deep Risk predictions within the last 24 hours (AUSPC)

Name	TPR vs PPV
8 hour classifier	0.790
12 hour classifier	0.771
16 hour classifier	0.760
20 hour classifier	0.711
24 hour classifier	0.664

However, in addition to the accuracy of the prediction, we are also interested in its *timeliness*: a correct prediction one minute before actual admission to ICU would be of little use. If we restrict to a window of length 4 hours, we can issue a prediction 1 hour ahead using information from 3 hours in the past, or 3 hours ahead using information from 1 hour in the past – but we cannot issue a prediction more than 4 hours ahead. Hence to issue timely predictions, we must use use longer windows. We achieve more accurate predictions by using shorter windows and more timely predictions by using longer windows. Because we want predictions that are both accurate *and* timely, we combine the predictions of various windows. To do so we train an additional RNN which we call the *ensembler* to combine these predictions.

CHAPTER 5

Results

We conducted our experiments on the dataset of [AYH17] that provides records for a cohort of 6,000+ patients in a large medical center.[1] The patient population is heterogeneous with a wide variety of diagnoses. (Extensive details and statistics of the dataset can be found in [AYH17].)

As in [AYH17], we divided the dataset into training and testing subsets based on admission date. The training set comprises the 81.0% of the patients who were admitted before July 1, 2015; the testing set comprises the remaining 19.0% who were admitted after July 1, 2015.

For Table 5.1, we compute the highest risk score for each patient at any time during the hospital stay, use that risk score as an indicator that the patient will be admitted to ICU *at some (later) point* during the hospital stay, and plot the AUSPC for the TPR vs. PPV trade-off. The left hand side of Table 5.1 compares the performance of Deep Risk with currently used clinical risk scores; the right hand side compares with competing machine learning methods. As can be seen, Deep Risk provides an enormous improvement over all the competitors. The GRU-based RNN benchmark that is used is simply our classifier which instead of feeding it with the encoder output, we use the original dynamic features.

[1] We are grateful to the authors of [AYH17] for sharing with us both the datasets and their results.

Table 5.1: Predictions during the hospital stay

Name	TPR vs PPV	TPR vs FPR
SOFA	0.123	0.691
APACHE III	0.144	0.662
MEWS	0.158	0.720
Random Forest	0.175	0.774
Logistic Regression	0.209	0.809
Gradient Boosting	0.210	0.816
Deep Risk	0.245	0.933

One aspect of timeliness is making predictions in advance of the actual occurrence of the event; another is making predictions that the event will occur in a specific time frame (e.g. within the next h hours). Table 5.2 shows the performance of Deep Risk in making such predictions, made at a single time during the hospital stay. Because most risk scores cannot make such comparisons, we show only the performance of Deep Risk using the ensembler, the performance of Deep Risk using only an unweighted average of the predictions of individual windows, and the performance of a state-of-the-art GRU-based RNN.

The ensembler does marginally worse than the unweighted average for prediction 6 hours ahead, marginally better for prediction 8 hours ahead and significantly better 10 hours ahead; both the ensembler and the unweighted average do much better than the GRU-based RNN for prediction in all three time frames.

Table 5.2: Prediction for specific time frames (AUSPC)

Name	TPR vs PPV 6 hrs	TPR vs PPV 8 hrs	TPR vs PPV 10 hrs
Deep Risk - Ensemble	0.379	0.370	0.368
Deep Risk - Average	0.384	0.364	0.354
RNN - GRU	0.247	0.246	0.276

Table 5.3 shows the performance of Deep Risk for predictions in a specific time frame, but made at a single time within 24 hours of the actual decision (to admit to ICU or to discharge). For these predictions, the ensembler does significantly better than the unweighted average in all three time frames; again, both do much better than the GRU-based RNN in all three time frames.

Table 5.3: Prediction for specific time frames close to decision time (AUSPC)

Name	TPR vs PPV 6 hrs	TPR vs PPV 10 hrs	TPR vs PPV 14 hrs
Deep Risk - Ensemble	0.523	0.522	0.483
Deep Risk - Average	0.500	0.500	0.472
RNN - GRU	0.362	0.381	0.380

The superiority of Deep Risk in issuing timely predictions is perhaps best seen in Figure 5.1 which shows PPV's (holding TPR fixed at 50%) at various times before the actual decision time for Deep Risk, the risk score of [AYH16] and medical risk scores.

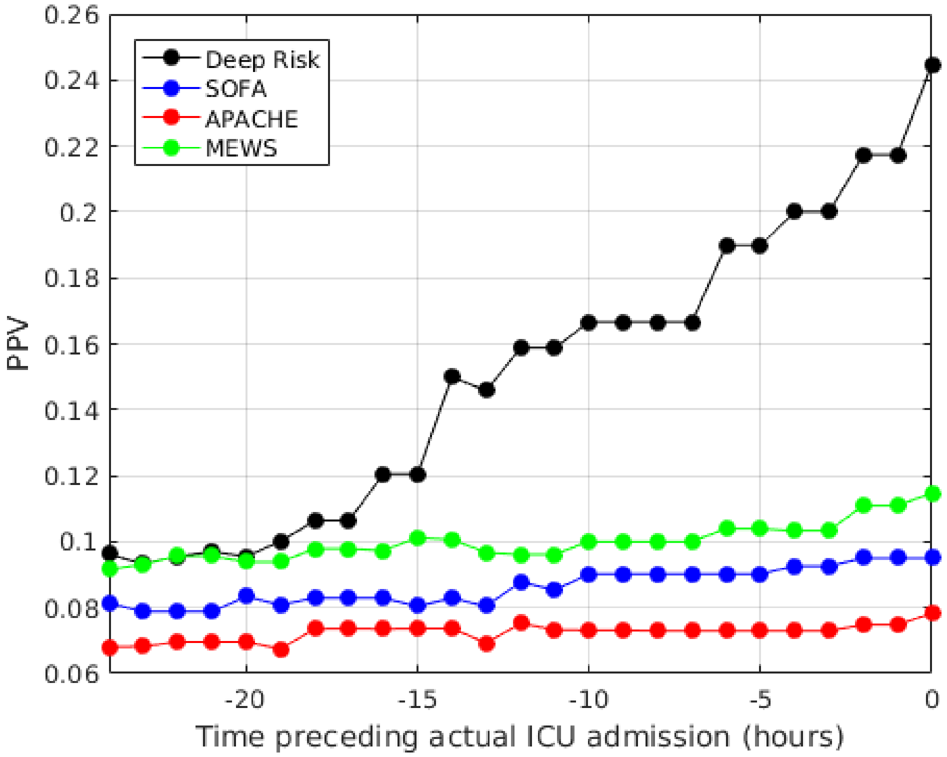

Figure 5.1: Prediction in the hours before decision: PPV holding TPR = 50%

Finally, we return to the effect of temporal regularization. In Table 5.4 we show the mean square error (MSE) in Deep Risk's reconstruction of the current input and the prediction performance (TPR vs PPV). Setting $\beta = \gamma = 0$ amounts to using a standard DAE trained jointly with the classifier. This has the effect of focusing on the current input and hence does a very good job of reconstructing the current input (achieves a low MSE) but at the cost of yielding relatively poor prediction. Perhaps surprisingly, setting $\alpha = \beta = \gamma = 0$ – i.e., not regularizing at all – does not do badly at prediction. (The decoder is not being trained and the encoder assumes the role of a simple additional GRU layer.) Performing full regularization (and choosing α, β, γ using cross-validation) does a good job of reconstructing the current input *and* yields the best predictions.

Table 5.4: The effect of regularization

Combination	MSE	TPR vs PPV
Full Regularization	0.350	0.873
No Regularization	1.173	0.847
$\beta = \gamma = 0$	0.287	0.817

CHAPTER 6

Conclusions

This paper has presented a Deep Learning architecture (Deep Risk) for predicting adverse events. The architecture of Deep Risk consists of two levels of RNNs: each of the lower-level RNNs makes a prediction based on information in its own specific time window; the higher-level RNN uses information from the longest of these time windows to aggregate those predictions into a single final prediction. Our methods enable Deep Risk to solve the key challenges inherent in the problem: non-stationarity of the data, the necessity of personalized and timely prediction, and unbalanced of data sets. To assess the performance of Deep Risk, we use it to make predictions about ICU admission, using a clinical dataset. Deep Risk achieves large and significant performance improvements over existing methods, including clinical risk scores and state-of-the art machine learning algorithms.

REFERENCES

[AYH16] Ahmed M Alaa, Jinsung Yoon, Scott Hu, and Mihaela Van der Schaar. "Personalized Risk Scoring for Critical Care Patients using Mixtures of Gaussian Process Experts." *ICML workshop on Computational Frameworks in Personalization*, 2016.

[AYH17] Ahmed M Alaa, Jinsung Yoon, Scott Hu, and Mihaela Van der Schaar. "Personalized Risk Scoring for Critical Care Prognosis using Mixtures of Gaussian Processes." *IEEE Transactions on Biomedical Engineering*, **PP**(99):1–1, 2017.

[CBS16] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. "Doctor ai: Predicting clinical events via recurrent neural networks." In *Machine Learning for Healthcare Conference*, pp. 301–318, 2016.

[CCP12] Lei Clifton, David A Clifton, Marco AF Pimentel, Peter J Watkinson, and Lionel Tarassenko. "Gaussian process regression in vital-sign early warning systems." In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 6161–6164. IEEE, 2012.

[CKL15] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. "Deep computational phenotyping." In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 507–516. ACM, 2015.

[HHP15] Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. "A targeted real-time early warning score (TREWScore) for septic shock." *Science Translational Medicine*, **7**(299):299ra122–299ra122, 2015.

[HV16] William Hoiles and Mihaela Van Der Schaar. "A Non-parametric Learning Method for Confidently Estimating Patient's Clinical State and Dynamics." In *Advances in Neural Information Processing Systems*, pp. 2020–2028, 2016.

[KCB15] David C Kale, Zhengping Che, Mohammad Taha Bahadori, Wenzhe Li, Yan Liu, and Randall Wetzel. "Causal phenotype discovery via deep networks." In *AMIA Annual Symposium Proceedings*, volume 2015, p. 677. American Medical Informatics Association, 2015.

[KDW85] William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. "APACHE II: a severity of disease classification system." *Critical care medicine*, **13**(10):818–829, 1985.

[PTP16] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. "Deepcare: A deep dynamic memory model for predictive medicine." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 30–41. Springer, 2016.

[RMS16] Narges Razavian, Jake Marcus, and David Sontag. "Multi-task Prediction of Disease Onsets from Longitudinal Laboratory Tests." In *Machine Learning for Healthcare Conference*, pp. 73–100, 2016.

[SKR01] CP Subbe, M Kruger, P Rutherford, and L Gemmel. "Validation of a modified Early Warning Score in medical admissions." *Qjm*, **94**(10):521–526, 2001.

[SR15] Takaya Saito and Marc Rehmsmeier. "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets." *PLOS ONE*, **10**(3):1–21, 03 2015.

[SS15] Peter Schulam and Suchi Saria. "A Framework for Individualizing Predictions of Disease Trajectories by Exploiting Multi-resolution Structure." In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pp. 748–756, Cambridge, MA, USA, 2015. MIT Press.

[YAH16] Jinsung Yoon, Ahmed M Alaa, Scott Hu, and Mihaela Van der Schaar. "ForecastICU: A prognostic decision support system for timely prediction of intensive care unit admission." In *International Conference on Machine Learning*, pp. 1680–1689, 2016.