A Study of Emergency Department Patient Admittance Predictors

Harish Kumar Manchukonda
Computer Science and Engineering
Mississippi State University
Starkville, Mississippi, USA
hm1089@msstate.edu

Shahram Rahimi
Computer Science and Engineering
Mississippi State University
Starkville, Mississippi, USA
sr2002@msstate.edu

Alexander Sommers
Computer Science and Engineering
Mississippi State University
Starkville, Mississippi, USA
ams1988@msstate.edu

Sean Bozorgzad
Health Technology Department
Potentia Analytics Inc.
Carbondale, Illinois, USA
sean@potentiaco.com

Junfeng Ma
Industrial and Systems Engineering
Mississippi State University
Starkville, MS, USA
ma@ise.msstate.edu

***Abstract*—We introduce and compare two prediction systems on the task of replicating human decisions regarding patient admittance in a typical American emergency department. The data-set used describes the patient trajectories in a 65,000 patient per-year emergency department in the United States. Among the descriptive attributes those of prime importance are the severity of the patient's condition and the time they waited to be admitted from the waiting room to the department proper. A recurrent neural network (RNN) is developed to learn the task of selecting the next patient from the waiting-room/queue to be admitted for treatment which is then compared to a heuristic-based selection algorithm currently used in industry for hospital simulation applications. We demonstrate achievable accuracies of 75.29% and 84.97% using the RNN, depending on the type of the data preprocessing used. These accuracies are only potentially and theoretically achievable, respectively. The former's validity hinges on whether certain "anomalous cases" are outliers or not, the second is achieved with the assumed existence of a method for labeling these same cases as anomalous as part of the RNN's input, which may or may not be achievable, pending further consultation with industry experts. Our conclusions hinge on whether or not such cases are outliers though in either case a more sophisticated data-set is desired. If they are not outliers then a more detailed data-set is likely necessary to apply machine learning, or at least our methods, meaningfully to this prediction problem for use in simulated, or real world, hospitals.**

***Index Terms*—Emergency Departments, RNN, Flow Optimization, Hospitals**

I. Introduction and Background

Emergency departments (EDs) are expensive and complicated systems critical to the mission of the hospitals they are embedded in. So much so that any process optimizations made can be pivotal to their successful operation at, or beyond, their previous levels. Before enforcing any changes in their processes or flows hospitals evaluate the effect of the considered modification using simulators [19]–[23]. The purpose of modeling and constructing an autonomous patient admittance predictor in this work is thus primarily to improve the quality of these simulations. Obviously the more realistic the components that make up the simulation the better the simulation represents the real world situation and the better will be the decisions made on the basis of that simulation. Our focus in this work is the component which selects the next patient to be admitted from the emergency department's waiting room. While the general rule of "most sever case first" is trivially simulated there are a number of deviations from such a pattern witnessed in normal hospital operation. These include "fast-track-beds" [10] and those caused by constraints in resource availability which, among others, can cause a naive algorithm to model real hospitals poorly. By examining this algorithm and constructing a new prediction system we hope to compare their usefulness for these simulations which are used to answer questions impacting operational efficiency, cost, and patient well-being. Potentially, a successful autonomous admittance system could be used beyond simulation to automate the patient admittance task, this could save money, free staff for other duties, and enforce consistency and fairness in admissions.

A. Emergency Department Flow

"An emergency department is a medical treatment facility specializing in emergency medicine and the acute care of patients who present without prior appointment; either by their own means or by that of an ambulance" [12].

A patient entering the ED will pass through 4 stages as described below and illustrated in Figure 1.

Stage 1, Triage: Triage assigns an emergency severity index (ESI) to each patient. This value ranges from 1-5 and indicates the severity of their condition with 1 being the most severe. Table I presents the meaning of each level.

Stage 2, Waiting to be treated: After being assigned an ESI the patient is admitted into the waiting room. Mathematically the waiting-room acts as a queue out of which patients are admitted into the ED proper. Ideally this queue is a priority queue where a patient's priority is computed from their ESI

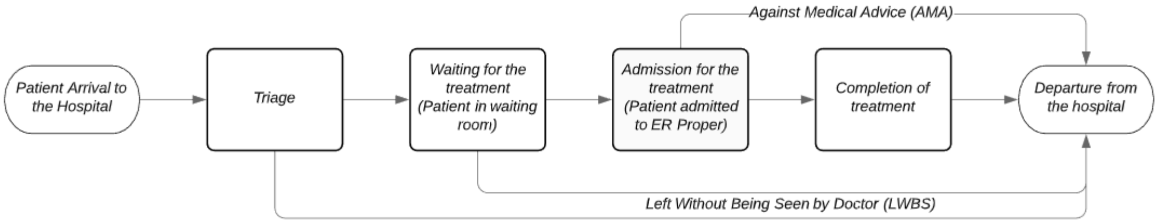


Fig. 1. Flowchart of ED visit

TABLE I
ESI LEVELS DESCRIPTIONS [25]

Level	***Description***
ESI 1	Patient requires immediate intervention to avoid death.
ESI 2	Emergency, patient is in a high risk condition, vitals are dangerously abnormal.
ESI 3	Urgent, multiple medical personnel are required to stabilize the patient but vitals are not dangerously abnormal.
ESI 4	Semi-Urgent, one staff member is required to stabilize the patient.
ESI 5	Non-Urgent, The patient is already in a stable condition.

and time already spent waiting. For patients of identical ESI and wait time the queue resolves these ties randomly. Deviations may occur from expected queue behavior due to human decision on the part of the patients (they might leave), staff, or on the basis of resource (medical personnel, beds, equipment, etc...) availability (a patient might not be admitted, even if they would be otherwise, due to the unavailability of a necessary resource, like a doctor specialized in their condition).

Stage 3, Admittance and treatment: Once a patient is admitted they are treated as necessary including them being assigned a bed, administration/prescription of drugs, and the performing of any needed medical imaging.

Stage 4, Departure: Once patients are able to return home they are discharged along with medical advice, equipment, and prescriptions to any needed medications. An alternative exit to the process comes when triaged patients leave the ED without being seen, a desertion most often due to wait times the patient deems unacceptable. A third transition to departure occurs when a fully admitted patient leaves before treatment is complete. These last two methods of departure are anomalous and unwanted events that, when they do occur, occur most often with low severity patients. Higher severity patients are understandably less likely/able to desert.

The contents of the rest of this paper are as follows: Section II talks about this area's related work, section III about our dataset and related methodology, section IV contains the results of our tests and a discussion of them, and, lastly, section V contains our future work.

II. RELATED WORK

Many papers [1]–[5] suggest that an ED can be modeled using discrete event simulation techniques. Discrete event simulation systems operate as a sequence of events across time where each event occurs at a particular time and on occurring the system marks the change.

Most of the work we found in this area was conducted in the simulation of an entire hospital, or one of its departments, to analyze flaws and anomalies such as long wait times and the associated patient desertion. [6] presents a hospital simulation tool *MedModel* which allows its users, hospital directors, to examine the complex operational and planning issues that emerge from the interaction of all the hospital's subsystems. [6], [7] present more focused simulation models, rare in that most simulations simulate entire hospitals with less detail rather than a detailed view of one department. The simulation is of all events (an event being any time a patient has something done to them in the system: they are admitted, they are treated, they are discharged, etc...) in the ED of a specific hospital named "The Cooper Health System" with the goal of reducing of the total length of the stay of each patient in the ED. [8] performed a functional analysis of a simulated hospital and discovered that most of a patient's time in the institution is spent waiting and proposed an operational procedure to reduce the waiting times in all scenarios. [9] presents a method of improving the patient flow in emergency departments by employing a dynamic priority queue, circumventing problems with FIFO, LIFO, and static priority queues, and suggesting the use of a dynamic priority "M/M/c" queue instead. [10] discussed the practice of emergency departments introducing fast-tracks-beds to improve patient safety and reduce waiting times. [11] surveyed queuing theory applications on health care systems and reported that congestion in the waiting-rooms happens when there is "poor quality of service" in the hospital. In [14] we presented a potential solution to the desertion phenomenon (itself called: "left without being seen by doctor" in industry) that becomes increasingly common when a patient waits a long time before being admitted. When a patient leaves like this it is a loss for the both parties: the patient isn't treated and the hospital losses potential revenue. Each patient's "leaving-probability" is calculated according to their waiting times and ESI level, similar to how their priority in the queue is calculated. When the leaving-probability of the patient reaches around 80%, their priority in the queue increases and they are informed.

III. Dataset and Methodology

The dataset used in this work contained around 65,000 records, though of them only 39,130 records presented data describing queues containing more than one patient (those describing single person queues were removed since a queue that only ever consists of one person, where that person is then called in, does not require prediction) and was collected at a private US hospital over a period of one year. The attributes of a patient in it are patient-ID, age, sex, ESI level, arrival time of the patient, length of time between arrival in waiting room (AIWR) and admittance to ED proper, length of time between AIWR and termination of treatment, length of time between AIWR and checkout from hospital, departure time of the patient, and total length of the stay of the patient in the hospital.

This data-set allows us to reconstruct a waiting-room queue as it evolves across a given day and we can use the time a patient waited before being admitted to the ED proper, and their ESI level, to calculate a derived value, the probability a patient will be called at the end of their x^{th} waited minute. This metric, a patient's "calling-probability", is produced from the data-set using a random forest as a function approximator. The function in question can be seen in Equation 1 but only works for combinations of ESI and waiting time extant in the data-set. Given each patient's calling-probability we can compare those patients which have the highest calling-probability in any given queue with the patients that were, in fact, admitted out of that queue, and see how closely they map to one another. This work concerns the use of the above described data-set to train and/or test systems which predict which patient is called into the ED from the waiting-room at any given calling occurrence.

$$P(C_{L,W}) = \frac{c_{L,W}}{(c_{L,W} + nc_{L,W})} \tag{1}$$

$\mathrm{P(C_{L,W})}$ (the calling probability) for a patient of ESI level L, with a wait time of W, equals $\mathrm{c_{L,W}}$ (a count of observed patients of that ESI level and w-time that were called in on their W^{th} waited minute) divided by the total number of patients witnessed in the dataset with those same characteristics whether called in ($\mathrm{c_{L,W}}$) or not ($\mathrm{nc_{L,W}}$) on the W^{th} minute. Calling-probabilities are specific to ESI level and are calculated as though a patient's likelihood of being admitted is independent of the other patients in the queue. Figure 3 shows the plotted calling-probabilities against time after AIWR for ESI levels 1 (crosses) and 5 (circles).

Because we needed a continuous function and our formula only works for combinations of ESI and wait-time extant in our dataset we needed to use a function approximator and selected a random forest as described below. The approximated function was then used to calculate calling-probabilities. The curves in Figure 3 do look similar to log curves suggesting the application of logistic regression when trying to model them, however this is more suitable for classification problems than regression problems [15].

For the uninitiated: "A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the data-set and uses averaging across each to improve predictive accuracy and control over-fitting." [13] To build our random forest we used the sklearn library version 3.2.4.3.2 and achieved an accuracy, averaged over all ESI levels, of 96%. During training we used a test-train-split of 80% train and 20% test, used a max depth of 3, and set the number of estimators (Number of trees in the forest) to 10. The models were then exported and were the functions used to calculate the calling-probabilities for the data-set as it was transformed.

A. Conversion of the data-set to the required form

In reference to Figure 2 the transformation of the data-set from its original form to one acceptable for training the RNN was straightforward.

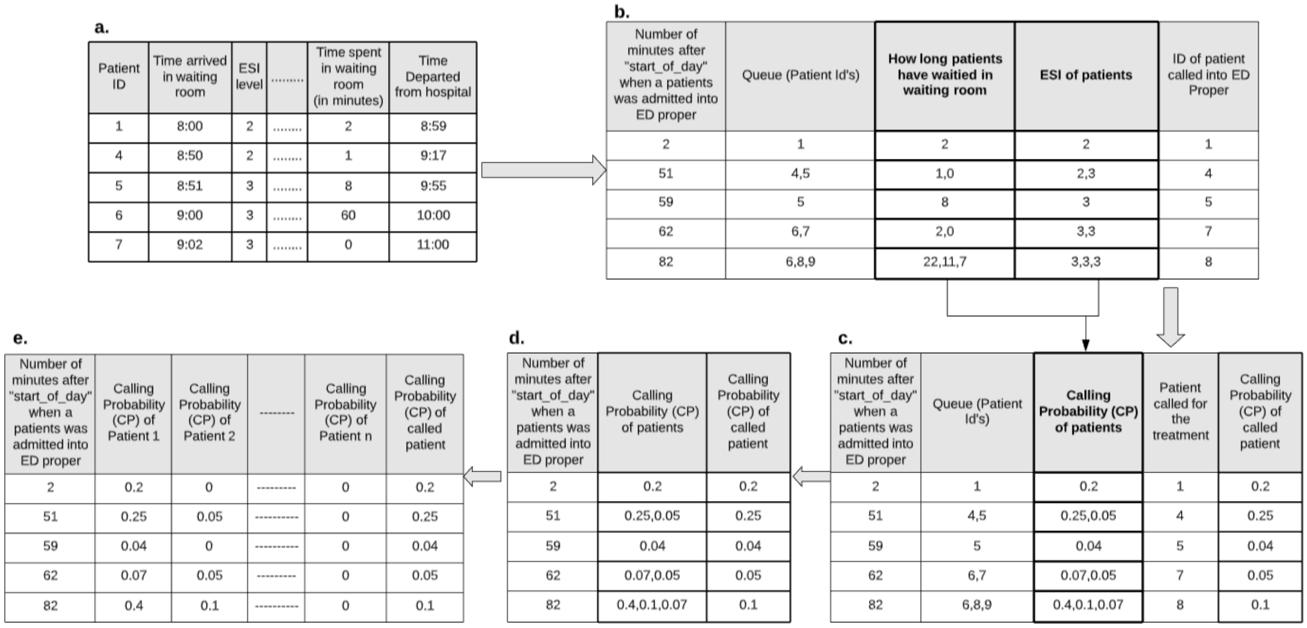
a) This is the original patient centered form of the data. Each record describes a patient's visit to the ED with their ID, a time-stamp of when they arrived in the waiting-room, their ESI level, etc...

b) From a. the queues are reconstructed and in the queue reconstruction form each record documents an instance of a patient in the waiting-room/queue being called and admitted into the ED proper. The values of how long a patient waited in the waiting-room, and what their ESI level was, are combined in form c via the function approximated by our random forest. It should be noted that the first attribute indicates the time since the "start-of-day" that has elapsed when a patient was called from the queue. Thus, if the start-of-day is 8:00am and a patient is called out of the queue at 9:02am, the entry for this attribute would be 62 minutes.

c) The calling-probability form of the data introduces the calling-probabilities which are ultimately used to judge and/or calculate the patient admission predictions that the prediction systems we tested make.

d) We drop attributes which are extraneous to the neural network and then expand the data in e., achieving the final form.

e) The neural-network form of the data is that which our neural net accepts as input. Each record is one labeled input that the neural net can be trained or tested with. Since our neural net required fixed length inputs the variable nature of the size of the queue is expressed in occupation of a queue with an upper limit of n patients where n = 49. Each non-zero calling-probability in the array represents one patient in the queue while absences in queues below 49 patients are indicated as 0. The label for the input (one input is one queue) is not a patient ID but instead a calling probability indicating that the selection, by a prediction system, of any patient with that calling probability would be a "correct" prediction.



a.

Patient ID	Time arrived in waiting room	ESI level	....	Time spent in waiting room (in minutes)	Time Departed from hospital
1	8:00	2	....	2	8:59
4	8:50	2	....	1	9:17
5	8:51	3	....	8	9:55
6	9:00	3	....	60	10:00
7	9:02	3	....	0	11:00

b.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Queue (Patient Id's)	**How long patients have waitied in waiting room**	**ESI of patients**	ID of patient called into ED Proper
2	1	2	2	1
51	4,5	1,0	2,3	4
59	5	8	3	5
62	6,7	2,0	3,3	7
82	6,8,9	22,11,7	3,3,3	8

c.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Queue (Patient Id's)	**Calling Probability (CP) of patients**	Patient called for the treatment	Calling Probability (CP) of called patient
2	1	0.2	1	0.2
51	4,5	0.25,0.05	4	0.25
59	5	0.04	5	0.04
62	6,7	0.07,0.05	7	0.05
82	6,8,9	0.4,0.1,0.07	8	0.1

d.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Calling Probability (CP) of patients	Calling Probability (CP) of called patient
2	0.2	0.2
51	0.25,0.05	0.25
59	0.04	0.04
62	0.07,0.05	0.05
82	0.4,0.1,0.07	0.1

e.

Number of minutes after "start_of_day" when a patients was admitted into ED proper	Calling Probability (CP) of Patient 1	Calling Probability (CP) of Patient 2	--------	Calling Probability (CP) of Patient n	Calling Probability (CP) of called patient
2	0.2	0	---------	0	0.2
51	0.25	0.05	----------	0	0.25
59	0.04	0	----------	0	0.04
62	0.07	0.05	---------	0	0.05
82	0.4	0.1	----------	0	0.1

Fig. 2. Conversion of original data-set to records of queues

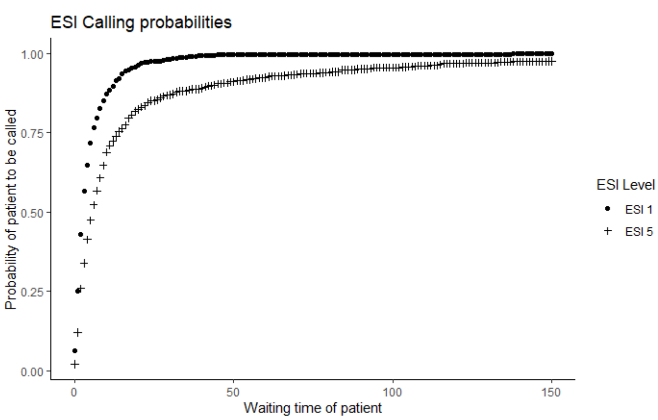


Fig. 3. Calling-probabilities against time for ESI’s 1 5.

B. Implementation

The near “pick-most-severe” algorithm: This algorithm, drawn from industry used simulations, nearly implements the obvious heuristic of picking the patient with the lowest ESI first but mildly modifies it with wait time considerations. Increasingly large weights, scaling up as ESI falls, are multiplied by each patient’s wait time to produce a score and the patient with the highest score gets selected. The weights on the lowest ESI levels are so large as to dwarf the effect of even the longest reasonable waiting times for higher ESI levels.

The recurrent neural network: Our RNN was trained on the transformed version of the data-set described earlier and was implemented using the Keras 2.3.0 library [17]. The training had a train-validation-test split of 64% training, 16% validation, and 20% testing and was trained for 100 epochs with a batch size of 100. The loss function was “means squared error” and the optimiser was the popular adam optimiser [18], The architecture consisted of 4 computation layers, the first being a “long short-term memory” (LSTM) layer with the rest being densely connected layers. A 25% dropout was used between layers one and two and layers two and three. The activation function used was the hyperbolic tangent function (tanh) because of the nature of our input. In order to retain our input’s expected dimensions we almost always had to pad them with 0s (padding was needed when there were fewer than 49 people in the queue) and we needed these null value’s effects to be propagated forward. Tanh, unlike sigmoid or softmax, allows 0 as a neural value. Our layer dimensions, starting from input and ending in output, are, in order: 49, 130, 65, 32, 16, 1, as seen in Figure 4. While doing so did not meaningfully changing the accuracy achieved we did achieve a 50% speedup in training using a CuDNNLSTM layer, a CUDA implemented version of an LSTM layer able to be run using a GPU [24].

IV. Results and Discussion

The tests performed to examine the success of each prediction system are straightforward. For each queue, at each step of it’s evolution, each system’s prediction of which patient is admitted (predicted patient) is compared to which patient was actually admitted (actual patient), as seen in the data, which is considered our “expected/accurate” choice. The industry algorithm achieved a baseline accuracy of 44.04%, meaning it replicated 44.04% of the choices a human made. In several cases the calling probabilities of the actual and predicted patients were rather close and if we broaden the notion of accuracy for “agreement” between actual and predicted patient we can see that accuracy, naturally, improves.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_7 (LSTM)                (None, 1, 130)            93600
_________________________________________________________________
dropout_13 (Dropout)         (None, 1, 130)            0
_________________________________________________________________
dense_26 (Dense)             (None, 1, 65)             8515
_________________________________________________________________
dropout_14 (Dropout)         (None, 1, 65)             0
_________________________________________________________________
dense_27 (Dense)             (None, 1, 32)             2112
_________________________________________________________________
dense_28 (Dense)             (None, 1, 16)             528
_________________________________________________________________
dense_29 (Dense)             (None, 1, 1)              17
=================================================================
Total params: 104,772
Trainable params: 104,772
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4. Neural Network Architecture(Generated by Keras)

TABLE II
ACCURACY MEASUREMENTS WITH ALGORITHM

Diff in calling-probability of actual and predicted patients deemed "accurate"	*Accuracy of the algorithm's predictions per level of acceptable error*
== 0.0	$\sim$ 40%
$\leq$=0.1	$\sim$ 49%
$\leq$=0.2	$\sim$ 58%
$\leq$=0.3	$\sim$ 65%
$\leq$=0.5	$\sim$ 80%

Table II shows the accuracy archived by the algorithm as the allowed difference in calling probability between the actual and predicted patient is increased. The allowed difference is the number of percentage points the actual and predicted patients' calling probabilities are allowed to differ by and still have the prediction count as accurate. In the first row, with the intuitive level of no allowed difference, the algorithm is only accurate if it selects the actual patient or one with an identical calling-probability. An accuracy of 80% is achievable but only at the cost of widening the notion of accuracy by an unacceptable 50 percentage points. All such results are achieved on the raw data-set, which is, of course, formed for neural input but has none of the so called "anomalous cases" removed.

The neural net, which is trained on this same data-set, and variants thereof (which will be explained), performs, with a return to a strict definition of an accurate choice, at a level of 44.15% accuracy (averaged across the results of a 5-fold cross validation), which is hardly an improvement on the industry algorithm. This suggested to us, at the point when we got that result, that the algorithm's poor performance might not be due to its simplicity.

What we mean by anomalous cases needs to be explained before the rest of our results are presented. What we call an anomalous case is any admittance witnessed in the data-set where there is at least a 50 percentage point difference between the calling probability of the actually admitted patient and the patient predicted to be admitted by the industry algorithm. The fundamental identity of these anomalous cases is unknown to us as the data-set does not characterize them enough to describe them meaningfully. We do not know how many varieties there are, what conditions in the hospital produce them, and indeed they are known to us in only a mathematical sense, and an assumed one at that, constructed to investigate our models' largest sources of error. We suspect these cases, based on conversation with an industry expert, to be made up of fast-track-bed cases or those admission anomalies caused by the availability of medical staff or other resources optimized for treating a patient causing that patient to be admitted earlier or later than was predictable. A variant of the data-set which removes these anomalous cases allows the RNN to perform at an accuracy of 75.29%, demonstrating a significant amount of the network's error came from them but that a significant amount more remains. The thus removed cases amount to 20.5% of the data-set and cannot, we originally supposed, be dismissed as outliers, at least en masse, though an industry expert assured us that they can be considered as such. The industry algorithm achieved an accuracy on this reduced data-set of 47.01% demonstrating that it is indeed sub-optimal, at least when tested with this data-set, even in the absence of obviously tough to predict outcomes.

Comparing these two results we can tentatively advance the RNN as a superior prediction system, not that it is surprising that the more expressive neural network is able to outperform a heuristic. If we assume all anomalous cases have an explanation (one was a fast-track-bed case, another was a burn case and the staff specializing in burns had nothing else to do just then so the patient was admitted very early, etc...) which could somehow be made present/indicated in the data a predicting system receives as input then we can assume that a more sophisticated data-set with these signals would allow the NN to perform better on the whole data-set than the industry algorithm simply by perceiving and computing on these additional characteristics. Lacking knowledge of what exactly these anomalous cases were we augmented our data by simply adding a binary flag to each input indicating whether it was or was not input for an anomalous case. This boosted the RNN's accuracy to 84.97% which is interesting in that such a performance boost was achieved without actually being able to characterize any anomalous case but by its presence. This suggests to us that perhaps even something as simple as one new input flag, if producible as part of a patient's triage, perhaps by some kind of system, or personnel, aware of resource availability issues at the time of their admittance, might be enough to let a NN achieve usefulness in simulation, and perhaps outside of it. Unfortunately it was beyond the scope of this work to attempt to interrogate the NN to learn how it was using this additional information.

Since our only way of detecting anomalous cases is an after-the-fact (anomalous cases are only known after a choice has been made by the industry algorithm and which can be compared to a reality which seems anomalous) significant discrepancy between the algorithmically and actually selected patients we cannot, at this time, produce a system that performs on real world data at more than an accuracy of 44.15%. However if, as we were told, the anomalous cases do indeed represent outliers, we can report a system which performs at the much more useful 75.29%; much depends on whether the anomalous cases can indeed be considered outliers. Pursuing a more expressive data-set, and reevaluating our methods using it, is part of our future work.

TABLE III
ACCURACY AND ERROR METRICS FOR THE RNN

Data Category	***R2***	***RMSE***	***MSE***	***MAE***
Raw dataset	44.15%	0.231094	0.053404	0.162643
Anomalous cases removed	75.29%	0.122312	0.014960	0.081476
Anomalous cases flagged	84.97%	0.106931	0.011434	0.058311

Table III presents the performance of the neural network in the form of the averaged r2 (accuracy) scores, mean square errors, root mean square errors, and mean absolute errors. Given the failure to perform well on the raw data-set these results are presented not as a useful measure of what can be practically achieved on real world data but to demonstrate the superior performance of the RNN when compared to the industry algorithm, especially if the anomalous cases are indeed to be disregarded. The last data-set, again, flags anomalous cases and returns them to the data-set. The reader should consider the results on the last data-set only theoretically achievable with a data-set more descriptive of the patients and the circumstances within the hospital, including a focus on available resources. Figure 5 shows the drop in loss as the training progressed for each data-set variant. What are we to make

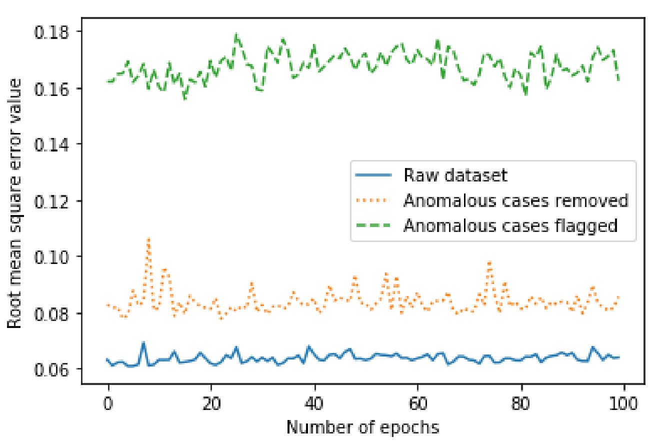


Fig. 5. Root mean squared error per data-set variant across epochs

of these results? Clearly the naive algorithm insufficiently models the real world data (though this does not mean it's guiding rule is itself flawed, we do not comment on the relative superiority of its rule or whatever rules the real world data may be the result of). The RNN performs well enough to be a step up as a part of a simulator but only if the problematically uncharacterized anomalous cases are eliminated, which may or may not be unrealistic. Though the confidential nature of healthcare data may be an impediment to such work we feel that it would be possible to achieve greater accuracy if more information about what we generically call "anomalous" cases were known, and not in the purely mathematical way we detect them now. Further, there is a wide gulf between a difference of 50 percentage points (the threshold between the calling probabilities of predicted and actual patient we take to be indicative of an anomalous case) and a difference of 0. Characterising that gap, especially when the difference rises beyond +- 10 percentage points, is likely necessary to improve accuracy further, likely in the form of more detailed data. We hope to continue our work on this data-set's more detailed decedents.

That the RNN performs as well as it does (a 2x boost to accuracy) when the anomalous cases are flagged is intriguing and suggests there is some rule yet to be recognized by us which could be of use even in the presence of the barest indication of a patient destined to be admitted anomalously.

V. FUTURE WORK

Our current plan is to consult subject matter experts, first to attempt to characterize the anomalous cases and then to see if we can be given an even mildly augmented data-set marking them (assuming one is producible). Another possible step forward is to make a patient's calling-probability calculated in a way so that it is considered dependent on the other occupants of the waiting room at that time. Outside of ED queue work we can construct further components to be added to the simulations used in research. One such project concerns the prediction of rates of patient arrival to the ED based on the current weather, the season of the year, and any recent public holidays. This work could facilitate better scheduling of medical staff and resources to cope with rises and falls in demand.

REFERENCES

[1] A. Komashie and Ali Mousavi, "Modeling emergency departments using discrete event simulation techniques"Proceedings of the Winter Simulation Conference,2005., Orlando, FL, 2005, pp. 5 pp.doi: 10.1109/WSC.2005.1574570

[2] Pajouh, Foad Mahdavi and Kamath, Manjunath,"Applications of Queueing Models in Hospitals" (2010). MWAIS 2010 Proceedings. 23. https://aisel.aisnet.org/mwais2010/23

[3] Samaha, Armel and Starks, "The use of simulation to reduce the length of stay in an emergency department," Proceedings of the 2003 Winter Simulation Conference, 2003., New Orleans, LA, USA, 2003, pp. 1907-1911 vol.2. doi: 10.1109/WSC.2003.1261652

[4] M. M. Gunal and M. Pidd, "Understanding Accident and Emergency Department Performance using Simulation," Proceedings of the 2006 Winter Simulation Conference, Monterey, CA, 2006, pp. 446-452.doi: 10.1109/WSC.2006.323114

[5] A. M. Hay, E. C. Valentin and R. A. Bijlsma,"Modeling Emergency Care in Hospitals: A Paradox - The Patient Should not Drive the Process," Proceedings of the 2006 Winter Simulation Conference, Monterey, CA, 2006, pp. 439-445.doi: 10.1109/WSC.2006.323113

[6] C. R. Harrell, C. R. Harrell, V. Lange, "Healthcare simulation modeling and optimization using MedModel," Proceedings of the 33nd conference on Winter simulation,https://dl.acm.org/doi/10.5555/564124.564155

[7] D. S. Technion, D. Sinreich, Y. N. Marmor, “A simple and intuitive simulation tool for analyzing emergency department operations,” Proceedings of the 36th conference on Winter simulation, 01-Dec-2004. https://dl.acm.org/doi/10.5555/1161734.1162107
[8] S. Takakuwa and H. Shiozaki, “Functional analysis for operating emergency department of a general hospital,” Proceedings of the 2004 Winter Simulation Conference, 2004., Washington, DC,USA, 2004, pp. 2003-2011 vol.2. doi: 10.1109/WSC.2004.1371562
[9] K. W. Tan, C. Wang and H. C. Lau, “Improving patient flow in emergency department through dynamic priority queue,” 2012 IEEE International Conference on Automation Science and Engineering (CASE), Seoul, 2012, pp. 125-130. doi: 10.1109/CoASE.2012.6386409
[10] Roche, K.T., Cochran, J.K. and Fulton, I.A. (2007) Improving patient safety by maximizing fast-track benefits in the emergency department – a queuing network approach, in Proceedings of the 2007 Industrial Engineering Research Conference, pp 619-624.
[11] Samuel, Herrmann, and J. W., “A Survey of Queuing Theory Applications in Healthcare,” Digital Repository at the University of Maryland, 26-Sep-2007. [Online]. http://hdl.handle.net/1903/7222
[12] “Emergency department,” Wikipedia, 04-Dec-2019. [Online]. Available: https://en.wikipedia.org/wiki/Emergency_department.
[13] “3.2.4.3.2. sklearn.ensemble.RandomForestRegressor” scikit. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.
[14] Harish Kumar, Shahram Rahimi,.. “Predictive Analytics for Left Without Treatment in the Emergency Departments”, 5th International Conference on Health Informatics Medical Systems (HIMS’19)
[15] S. Narkhede, “Understanding Logistic Regression,” Medium, 26-May-2019. Available: https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102
[16] “running Python functions as pipeline jobs,” Joblib. https://joblib.readthedocs.io/en/latest/
[17] “Keras: The Python Deep Learning library,” Home - Keras Documentation. [Online]. Available: https://keras.io/
[18] Optimizers - Keras Documentation. [Online]. Available: https://keras.io/optimizers/
[19] G. W. Evans, E. Unger and T. B. Gor, “A simulation model for evaluating personnel schedules in a hospital emergency department,” Proceedings Winter Simulation Conference, Coronado, California, USA, 1996, pp. 1205-1209.
[20] Paul, S. A., Reddy, M. C., DeFlitch, C. J. (2010). A Systematic Review of Simulation Studies Investigating Emergency Department Overcrowding. SIMULATION, 86(8–9), 559–571.
[21] “A Simulation Model for Scheduling in the Emergency Room,” International Journal of Operations Production Management.
[22] A. P. Kumar and R. Kapur, “Discrete simulation application-scheduling staff for the emergency room,” Discrete simulation application-scheduling staff for the emergency room — Proceedings of the 21st conference on Winter simulation, 01-Oct-1989.
[23] M. A. Centeno, R. Giachetti, R. Linn, A. M. Ismail, Martha A. “Emergency departments II: a simulation-ilp based tool for scheduling ER staff,” Emergency departments II — Proceedings of the 35th conference on Winter simulation: driving innovation, 01-Dec-2003.
[24] Recurrent Layers - Keras Documentation. [Online]. Available: https://keras.io/layers/recurrent/
[25] Tucci, V., Ahmed, S.M., David, oyer, R., Greene, S., Nidal, Moukaddam (2017). Stabilizing Intentional Overdoses in Freestanding Emergency Departments : A Good Idea ?