AI-Assisted Annotator Using Reinforcement Learning

V. Ratna Saripalli · Dibyajyoti Pati · Michael Potter · Gopal Avinash · Charles W. Anderson

Received: 25 July 2020 / Accepted: 28 September 2020 / Published online: 9 October 2020
© Springer Nature Singapore Pte Ltd 2020

Abstract
Machine learning in the healthcare domain is often hindered by data which are both noisy and lacking reliable ground truth labeling. Moreover, the cost of cleaning and annotating this data is significant since, unlike other data domains, medical data annotation requires the work of skilled medical professionals. In this work, we introduced the use of reinforcement learning to mimic the decision-making process of annotators for medical events allowing automation of annotation and labeling. Our reinforcement agent learns to annotate health monitor alarm data based on annotations done by an expert. We demonstrate the efficacy of our implementation on ICU critical alarm data sets. We evaluate our algorithm against standard supervised machine learning and deep learning methods. Compared to SVM and LSTM methods, our method achieves high sensitivity that is critical for alarm data; exhibits better generalization across mixed downsampling; and preserves comparable model performance.

Keywords False alarms · Reinforcement learning · Annotation

Introduction
Healthcare costs have increased significantly worldwide over the past decades, and especially within the United States where they now account for nearly 18% of the GDP. As such, it is imperative to find ways to lower systemic costs while not compromising quality of care. In particular, Operating Room (OR) costs represent a significant fraction of a hospital’s expenditures. Many sequential decision-making steps are involved in the day-to-day function of an OR. Common examples include deciding when to transfer patients to post-anesthesia care units, scheduling staff for care units, determining surgery end, estimating emergence phase, estimating time to extubate, and setting critical event alarms. These problems present attractive targets for novel machine learning applications which, leveraging an ever-growing corpus of medical data, may be able to significantly streamline hospital logistic operations.

Data are the fundamental currency for solving many healthcare problems using computational methods. While volumes of medical data are increasingly being made available, these datasets bring their own unique challenges. Medical data are plagued with concerns ranging from data privacy and ground truth availability, to domain issues such as sparsity and heterogeneity, to quality issues such as missing data and noise [1–4]. Moreover, obtaining trustworthy annotations of medical monitoring data is expensive and time consuming due to domain expertise requirements [6]. On the other hand, medical devices such as anesthesia machines, ventilators, and monitoring systems can be a rich source of data as they assist in processing, identifying, and generating alerts for patients. Many such alerts in turn serve as the basis for optimal OR decision-making.

One well-known problem with alerts from conventional medical monitoring systems is alarm fatigue, which can be attributed to the threshold-based alarm classification approaches currently used in such systems [5, 6]. Patients and their families become anxious when alarms indicate a
change in the patients’ health and a need for medical attention. An overabundance of false alarms also tends to drive operators to either silence or ignore the alarms—reducing the utility of the monitoring device. To build smarter alarm systems leveraging modern deep learning algorithms, we must address the problem of false alarms. This requires a significant volume of correctly annotated data. Unfortunately, as noted before, this is both expensive and time consuming [6].

Reinforcement learning (RL) is a computational approach to learning from interactions that is goal focused and has gained a lot of attention in the last 5 years [7]. RL algorithms center around an agent that senses, observes, and interacts with an environment. The environment, in turn, either rewards or penalizes the agent to attain a specific goal. RL is especially useful in the automation of tasks which require human goal-oriented action and sequential decision-making. These algorithms have seen great success even in complicated tasks such as playing games at human and super-human levels of performance [8]. We report on a novel application of RL to the domain of medical alarm annotation. In particular, we compare several RL algorithms and feedback mechanisms in an effort to mimic a human expert at the task.

Related Work

Electronic Health Record and Electronic Medical Record health systems have matured over the past decades, and yet the data generated from these systems are yet to be tapped to their full potential. Many traditional approaches have been used for detecting false alarms that depend on feature engineering and the availability of ground truth (labeled) data [5, 9–13]. For example, Wang et al. [5] used a three-step approach of feature extraction, selection, and classification based on arterial blood pressure (ABP) and electrocardiogram (ECG) signals for false alarm detection. The authors found that direct raw signals yield poor results due to noise and unstable voltages. To overcome this, they developed a customized feature set using statistical methods, which was then passed to a support vector machine (SVM) to classify the alarms. While their work achieved high classification performance, the results are limited to ECG arrhythmia alarms and required significant feature engineering.

Although such traditional approaches to false alarm detection give reasonable model performance, they suffer from several significant limitations. One chief concern is that they typically focus on a single alarm/signal type (e.g., arrhythmia) and, therefore, cannot scale and generalize for various alarm types. Another important concern is their poor performance when only a small set of ground truth data is available. Distant supervision—mapping of entity relations from a known knowledge base to a dataset that has unlabeled data [21]—can be used to alleviate the limited ground truth problem, but such methods still do not address the generalization problem [6]. In their work, Schwab et al. [6] use a multitask network architecture to perform auxiliary tasks and detect false alarms via distant supervision. While this proved to be beneficial when the available labeled dataset was small (under 100 samples), their work was limited to only detecting false alarms due to technical errors or artifacts. It also required the use of auxiliary tasks, which is not always natural or desirable for a given dataset.

Moving beyond classical methods, RL algorithms have begun to show success in medical domain problems such as defining ventilation weaning protocols and customizing drug administration strategies [14–17]. In 2015, the field of deep RL was born when researchers combined RL techniques with deep neural networks to create the Deep Q-Network (DQN). The DQN was able to solve complex state-space problems in the form of Atari 2600 games, at the level of a professional human player, without any prior domain knowledge [8]. A more recent RL algorithm, the advantage actor-critic (A2C), learns to approximate both its policy and value functions and then uses the latter to update the former [7, 19]. We see the benefits of A2C’s learned value function as compared to the DQN approach in our results (Sect. 4).

Methods

We trained DQN and A2C agents to annotate medical signal data based on whether or not it represents an alarm state. The high level data flow is diagrammed in Fig. 1.

Our proposed RL approach can learn from the decision-making of the domain expert without any assumptions regarding signal type nor pre-encoding any domain expertise. Once the RL agent reaches reasonable performance, we can replace the human expert with the RL agent to annotate the data. Depending on accuracy requirements, a production system might leave a human in the loop to validate the annotations output by the RL agent, but their workflow could be accelerated by access to the agent predictions.

Datasets

We used the multiphasic Push Electronic Relay for Smart Alarms for End User Situational Awareness (PERSEUS) program’s data hosted by Brown University’s digital archive [22]. These data were generated from an adult Emergency Department (ED) for a regional referral medical facility and level I trauma center. It contains data from patient monitoring devices within a 15-bed urgent care area in the ED. The PERSEUS dataset contains 12 months of.json formatted data. The data are split across files by
monitoring device and by 24 h increments. The following signals are recorded in each file:

- Electrocardiogram waveform (single lead EKG, Lead II) at 250 Hz.
- Pulse oximetry waveform (PPG) at 125 Hz.
- Vital signs: heart rate (HR), respiratory rate (RR), systolic blood pressure (SBP), diastolic blood pressure (DBP), mean arterial blood pressure (MAP) and peripheral capillary oxygen saturation (SpO₂).
- Alarm messages (institution-specified alarms).

Kobayashi et al., as part of the PERSEUS program, developed several data subsets with annotation for experimental (non-clinical) research known as adjudicated/annotated telemetry signals for medically important and clinically significant events (ATOMICS), which are used in this research [20]. From the 12 months of PERSEUS data, three non-consecutive weeks of red alarms data are annotated by Kobayashi et al. for clinical significance and severity to form the ATOMICS dataset. This dataset is then further divided by weeks into ATOMICS-1, ATOMICS-2, and ATOMICS-3. We use ATOMICS-1 as our training data and ATOMICS-2 as our testing data.

### Preprocessing Data

We first convert all alarms and annotations to one-hot encodings for processing. The annotations are divided into two categories of actions (alarms/non-alarms) to simplify the problem space. The clinically significant and severe alarms (emergent, urgent) are categorized as alarms, while the indeterminate and non-urgent events are categorized as non-alarms. The raw data from three files—vitals, alarms, and annotations—are then merged together into a single file based on timestamps. This results in dataset 1 (DS1) as summarized in Table 2. We then create a second dataset (DS2) from DS1, where ‘indeterminate’ events are removed. These are events where the professional annotator was unable to make a definite assessment of alarm status. Two examples of such events can be seen in Fig. 2, where the indeterminate event can be seen to share some attributes of a true alarm and some attributes of a non-alarm. DS2 is also down-sampled from milliseconds to seconds using mean value imputation. Its final data distribution is summarized in Table 2.

The data are highly imbalanced and sparse for critical clinically significant alarm events. Each critical alarm is surrounded by 600 non-alarm events before and after. This results in an imbalance ratio of 1:1200. To rectify this imbalance, we used the following downsampling techniques: n-0, n-1, n-3, n-5, n-10, and mixed. In n-0 downsampling, we retain only alarm data. In n-1, we retain 1 non-alarm before
The goal of the RL agent is to maximize its expected reward using known examples to learn an optimal policy.

### Learning an Optimal Policy

Learning the best mapping (Q-function) between actions and states is the essence of reinforcement learning. We compare two different models for learning this mapping: a DQN with experience replay (value-based learning [8]), and an A2C (policy-based learning [7, 19]).

The DQN network takes in $s_t$ as described in Sect. 3.3, and it outputs a $Q$-value for each action $a \in A$ pursuant to Eq. 2. The model parameters ($\theta$) are updated after every 10 steps of training within each epoch, with a batch size of 8, learning rate of 0.001, and an Adam optimizer. The discount factor ($\gamma$) is set to 0.9.

$$\hat{Q}_k(s_t, a_t) \leftarrow r_{t+1} + \gamma \max_{a \in A} \hat{Q}_{k-1}(s_{t+1}, a, \theta).$$

(2)

The optimal policy $\pi^*$ for the DQN method after $k$ iterations is given in Eq. 3:

$$\pi^*(s) = \arg \max_{a \in A} \hat{Q}_k(s, a).$$

(3)

The A2C algorithm is comprised of two components: a policy function and a value function. To update these functions, A2C maintains two neural networks. The critic network predicts the value $V_w(s)$ at time $t$ from the state parameters $s_t$, and updates the value function parameters $w$, which in turn are used for computing ‘Advantage’ as described in Eq. 4.

$$\text{Advantage}(s_t, a_t) \leftarrow r_{t+1} + \gamma V_w(s_{t+1}) - V_w(s_t).$$

(4)

The value function for a given policy is described in Eq. 5:

$$V(s_{t+1}) = \mathbb{E} \left[ \sum_{t=1}^{T} \gamma^t r_t \right].$$

(5)

In Eqs. 4 and 5, $r_t$ is the reward after time $t$. The actor updates the policy per the critic to maximize ‘Advantage’ at a given time $t$ from the input state parameters $s_t$.

Both the actor and critic networks are updated every 10 time steps during an epoch, with a batch size of 8. The networks use an Adam optimizer with a learning rate of 0.001 for the actor and 0.005 for the critic. The discount factor ($\gamma$) is set to 0.9.

Both the DQN and A2C models choose actions based on a decaying $\epsilon$-greedy rule with an initial value of 1 annealed to 0.01 using a decay factor of 0.99975.
Experiment Design

All experiments were run on a MacBook Air with an Intel Core i5 1.8 GHz processor and 8 GB of RAM. Training was performed using ATOMICS-1 data, and all agent evaluations were conducted using ATOMICS-2 data. Table 2 summarizes the characteristics of the datasets. Standard binary classification metrics such as sensitivity, specificity, and the Mathews correlation coefficient (MCC) [18] were employed to compare model performance. Code was written using the Keras sequential model API and Python 3.6.

Benchmarking Against Non-RL Algorithms

We also benchmark our RL algorithms against three non-RL-based approaches: a multi-layer perceptron (MLP), an SVM, and a recurrent neural network using long short-term memory (LSTM). The MLP and SVM both take \( s_t \) as input and return a classification value in \{1, 0\} for alarms and non-alarms. They are benchmarked by training with n-0, n-3, and n-10 downsampling on DS1. The LSTM is fed a sequence of \( s_t \) values surrounding the target event, and then returns a classification value. If the event occurs at time \( t \), then an LSTM with window size \( w \) sees state data in the range \( s_{t-w/2} \rightarrow s_{t+w/2} \).

The MLP consists of two hidden layers with twenty and four nodes, respectively, each using a hyperbolic tangent activation function leading into a softmax output layer. It was trained until convergence for a fair comparison with the RL agent. For n-3 downsampling, the class weight was set to 20:1, and for n-10 downsampling, it was set to 40:1. Without adjusting these weights, the classifier collapses to always predicting the majority class.

We used scikit-learn’s sklearn.svm package, and support vector classifier (SVC) for our SVM experiments. After experimenting with poly, rbf, and linear kernels, we selected the linear kernel as this gave the maximum AUC score. We also set the ‘class weight’ parameter to ‘balanced’. Without adjusting these weights the classifier collapses to always predicting the majority class. All other parameters were left as their package default values.

Our recurrent neural network consists of four stacked layers with two initial LSTM layers followed by two dense layers. We used a batch size of 16, the RMSProp optimizer, and a learning rate of 0.001.

Results and Discussion

The purpose of an AI-assisted annotator is to mimic an expert annotators behavior. Our results are discussed in detail in the following sections. A2C agents have previously been found to learn sparse events more effectively than DQNs, a finding which is borne out here as well. The following sections detail the experimental results from this study.

Comparing the Adam and RMSProp Optimizers

The performance of various agents trained at different epochs was evaluated and compared for DQN and A2C using F1 scores weighted by class prevalence, as seen in Fig. 3. We used n-mixed downsampling on DS1.

As can be seen from Fig. 3, the Adam optimizer was found to be more stable than RMSProp. RMSProp performance fluctuated significantly for DQN and resulted in a worse final performance for both DQN and A2C. We, therefore, continued with Adam for the rest of the experiments.

Comparing the Results of A2C and DQN Agents

In medical alarm annotation, sensitivity is a very important metric since we do not want to miss any true alarms. Specificity is the ‘value-added’ metric which indicates how effectively we can identify false alarms. The performances of our A2C and DQN algorithms are compared with respect to these metrics in Table 3, along with AUC and MCC scores. Each agent (A2C and DQN) was run for 5000 epochs of training, being evaluated against the test dataset every 100 epochs. The top 3 (highest AUC) evaluation results during this training process are reported in Table 3. We find that A2C consistently achieves higher sensitivity scores than DQN, at best identifying 89% of true alarms compared with 76% in the case of DQN. On the other hand, that A2C network would catch only 42% of false alarms compared to the DQN’s 64%. While catching more false alarms would be nice, a realistic system should not miss any true alarms, so A2C is preferable.
A2C Agent Training Curves and Reward Signals

The training curve for an A2C agent is provided in Fig. 4. This captures the model's average accumulated reward per epoch when training with n-mixed downsampling on DS1. As can be seen from the figure, the network steadily learns a better action policy.

Benchmarking RL and ML Results

We benchmarked our top A2C agent results against MLP and SVM binary classification results with n-0, n-3, and n-10 downsampling on DS1 (see Table 2). The results of this benchmarking are presented in Table 4. In these tests the SVM mostly outperforms A2C and MLP. It is worth noting, however, that if the MLP and SVM are not adjusted for class imbalances they both collapse to majority-only classifiers. A2C has no explicit class re-balancing, so we also run n-mixed downsampling as a pseudo-imbalance rectifier. The result is that the A2C model achieves the highest recorded sensitivity while maintaining a 45% specificity.

Alternate Reward Function on DS2

In this section, we explore the performance of A2C when using a more sophisticated reward function (Eq. 1). These experiments used DS2 and were benchmarked against an LSTM. Results are given in Table 5. The A2C with access to only a single $s_t$ is able to outperform the LSTM with window size 20 by a notable margin. When the LSTM is given access to an even larger window size it gains the advantage on most performance metrics, but still fails to match the sensitivity of the A2C algorithm. This suggests a possibly interesting follow-up study where window data are provided to the A2C algorithm as well.

Conclusion

In this work, we design and demonstrate the application of deep reinforcement learning to the healthcare domain. Supervised and semi-supervised approaches to false alarm detection require feature engineering and domain expertise to scale and generalize, which is data intensive and
expensive. Our RL approach mimics medical domain experts with high sensitivity, while still catching a notable number of false alarms. To the best of our knowledge, this is the first application of deep RL for medical alarm annotation.

Agents trained with our mixed downsampling technique have superior performance compared with any single downsampling approach in this highly imbalanced data set. Our optimal A2C configuration achieved 88.5% sensitivity in detecting true alarms, and 45.2% specificity in identifying false alarms. This was accomplished after training on only 1 week’s worth of scanner data.

Identifying clinically false alarms, as done here, is a significantly more difficult task than identifying false alarms caused by artifacts and technical errors. Clinical reasoning requires an understanding of the patients’ high-level physiological state as well as significant domain knowledge. Expert annotators used 20 s windows of data to visualize and adjudicate the alarms. Our techniques were competitive with LSTMs which simulated that view of the data, while requiring only a single data point. We believe that expanding RL techniques to the broader data window could result in significantly improved performance. It may also be interesting to represent the signals visually before feeding them into a hybrid CNN/RL algorithm.

We would like to extend this work to specific alarm types (emergent, urgent, indeterminate) and refine the reward function in our future work. Our current work, however, can be applied to any event detection problem that follows a Markov decision process.

The limitations of this work are (i) training and testing of the RL agent are limited to 1 week of data; (ii) we could find no prior work employing deep RL techniques on false alarm detection to benchmark against; (iii) the alarm detection task is limited to two classes.

The main contributions of this work are (i) A2C performs and generalizes better than DQN; (ii) Adam optimizer is more stable than RMSProp for this problem; (iii) mixed sampling ranges perform well for RL state representation compared to uniform downsampling; (iv) our approach is data efficient, scalable to multiple tasks, and not very compute intensive. Furthermore, such methods could soon enable many practical non-clinical applications such as easing the data annotation processes thereby allowing faster generation of new healthcare datasets.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhou L, Pan S, Wang J, Vasilakos AV. Opportunities and challenges. Machine learning on big data. Neurocomputing. 2017;237:350–61.
2. Xiao C, Choi E, Sun J. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. In: JAMIA. 2018.
3. Ghassemi M, Naumann T, Schulam P, Beam AL, Ranganath R. Opportunities in machine learning for healthcare. 2018. arXiv:1806.00388.
4. Esteva A, Robicquet A, Ransdambad B, Kuleshov V, DePrisco M, Chou K, Cui C, Corrado G, Thrun S, Dean J. A guide to deep learning in healthcare. Nat Med. 2019;25(1):24.
5. Wang X, Gao Y, Lin J, Rangwala H, Mittu R. A machine learning approach to false alarm detection for critical arrhythmia alarms. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). IEEE; 2015. p. 202–207.
6. Schwap P, Keller E, Muroi C, Mack DJ, Strässle C, Karlen W. Not to cry wolf: Distantly supervised multitask learning in critical care. 2018. arXiv:1802.05027.
7. Sutton RS, Barto AG, et al. Introduction to reinforcement learning, vol. 2. Cambridge: MIT Press; 1998.
8. Volodymyr M, Koray K, David S, Andrei AR, Joel V. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529–33.
9. Sayodi O, Shamsollahi MB. Life-threatening arrhythmia verification in icu patients using the joint cardiovascular dynamical model and a Bayesian filter. IEEE Trans Biomed Eng. 2011;58(10):2748–57.
10. Clifford GD, Silva I, Moody B, Li Q, Kella D, Shahin A, Kooistra T, Perry D, Mark RG. The physionet/computing in cardiology challenge 2015: reducing false arrhythmia alarms in the icu. In: 2015 computing in cardiology conference (CinC). IEEE; 2015. p. 273–76.
11. Plesinger F, Klimes P, Halamek J,Jurak P. Taming of the monitors: reducing false alarms in intensive care units. Physiol Meas. 2016;37(8):1313.
12. Salas-Boni R, Bai Y, Harris PRE, Drew BJ, Hu X. False ventricular tachycardia alarm suppression in the icu based on the discrete wavelet transform in the ecg signal. J Electrocardiol. 2014;47(6):775–80.
13. Behar J, Oster J, Li Q, Clifford GD. ECG signal quality during arrhythmia and its application to false alarm reduction. IEEE Trans Biomed Eng. 2013;60(6):1660–6.
14. Prasad N, Cheng L-F, Chivers C, Draugelis M, Engelhardt BE. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. 2017. arXiv:1704.06300.
15. Escandell-Montero P, Chermsi M, Martiniz-Martinez JM, Gomez-Sanchis J, Barbier C, Soria-Olivas E, Mari F, Vilan-Francces J, Stopper A, Gatti E. Optimization of anemia treatment in hemodialysis patients via reinforcement learning. Artif Intell Med. 2014;62(1):47–60.
16. Nemati S, Ghassemi MM, Clifford GD. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach. In: 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE; 2016. p. 2978–81.
17. Padmanabhan R, Meskin N, Haddad WM. Closed-loop control of anesthesia and mean arterial pressure using reinforcement learning. Biomed Signal Process Control. 2015;22:54–64.
18. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genom. 2020;21(1):6.
19. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. 2016. p. 1928–37.

20. Kobayashi L, Oyalowo A, Agrawal U, Chen S-L, Asaad W, Hu X, Loparo KA, Jay GD, Merck DL. Development and deployment of an open, modular, near-real-time patient monitor datastream conduit toolkit to enable healthcare multimodal data fusion in a live emergency department setting for experimental bedside clinical informatics research. IEEE Sens Lett. 2018;3(1):1–4.

21. Mintz M, Bills S, Snow R, Jurafsky D. Distant supervision for relation extraction without labeled data. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, vol 2, no 2. 2009. p. 1003–11.

22. Kobayashi L, Oyalowo A, Agrawal U, Hu X, Loparo KA, Leary OP, Jay GD, Merck DL. Push electronic relay for smart alarms for end user situational awareness (PERSEUS) research program full original dataset. Brown University Open Data Collection. PERSEUS/ATOMICS Digital Archive. Brown Digital Repository: Brown University Library; 2018.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.