

---

# Predicting Adolescent Suicide Attempts with Neural Networks

---

**Harish S. Bhat**  
Applied Mathematics Unit  
University of California, Merced  
Merced, CA 95343  
hbhat@ucmerced.edu

**Sidra J. Goldman-Mellor**  
Public Health  
University of California, Merced  
Merced, CA 95343  
sgoldman-mellor@ucmerced.edu

## Abstract

Though suicide is a major public health problem in the US, machine learning methods are not commonly used to predict an individual's risk of attempting/committing suicide. In the present work, starting with an anonymized collection of electronic health records for 522,056 unique, California-resident adolescents, we develop neural network models to predict suicide attempts. We frame the problem as a binary classification problem in which we use a patient's data from 2006-2009 to predict either the presence (1) or absence (0) of a suicide attempt in 2010. After addressing issues such as severely imbalanced classes and the variable length of a patient's history, we build neural networks with depths varying from one to eight hidden layers. For test set observations with at least one prior diagnosis in the category of schizophrenia and psychotic disorders, our model achieves a sensitivity of 0.78, specificity of 0.91, AUC of 0.92, and overall accuracy of 88%. We detail these and similar results, and then pose open questions for future work.

## 1 Introduction

Recently, both suicide fatalities and attempts have increased among US adolescents. Between 2009 and 2014, suicide deaths among youths aged 10-19 increased 17%, and nonfatal suicide attempts that resulted in an emergency department (ED) visit soared 65% [2, 18]. At the same time, the predictive capability of clinicians and standard statistical models is only slightly better than chance [8]. Given the ever increasing volume of electronic health records (EHRs), it is natural to ask whether machine learning methods might be capable of improving the accuracy of suicide attempt prediction. If so, such methods might one day be incorporated into clinical tools that flag high-risk patients, potentially improving their lives.

Modern deep learning methods have not made their way into the mainstream suicide prediction literature. Recent work features methods such as random forests [23] and  $k$ -nearest neighbors [22]. Other studies involve special data sources, such as textual notes in EHRs [10], survey data [6, 7, 20, 25], or professional risk assessments [22]. This enriched data enables the use of methods from, e.g., natural language processing and ordinal regression. When neural network methods have been used for suicide prediction, they have performed well [11, 15]. Outside of suicide research, deep learning has enjoyed recent success in the medical informatics literature [3, 19], further motivating the present work.

## 2 Data

This study was approved by (our university's) Institutional Review Board. We used nonpublic versions of California emergency department encounter and hospital admissions data from 2006 through

2010. The California Office of Statewide Health Planning and Development (OSHPD) provided anonymized individual-level patient encounter data from all California-licensed hospital facilities, including general acute care, acute psychiatric, chemical dependency recovery, and psychiatric health facilities, but excluding federal hospitals. ED and inpatient data were screened by the OSHPD’s automated data entry and reporting software program (MIRCal); data fields with error rates of 0.1% or higher were returned to the hospitals for correction [16, 17]. Patients with missing age were excluded.

The study dataset consists of ED and inpatient records for all adolescent patients aged 10 to 19 years who had a valid unique identifier (encrypted social security number or SSN) and a California residential zip code in 2010 (64.0% of all records for this age group). Unique identifiers were used to link multiple ED visits/episodes per patient over time, including encounters prior (2006-2009) to the adolescent’s first recorded 2010 visit. A total of 522,056 unique, CA-resident adolescents are available for these analyses. Of these, 5,490 adolescent patients presented in 2010 to an ED with a suicide attempt code, i.e., a primary International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM [14]) External Cause of Injury code (E-code) of E950.0-959.x [5].

Note that prior neural network suicide prediction models were trained using data from, respectively, 91 and 7399 patients, nearly 70 times less than the present work [11, 15]. Significantly, these and many other machine learning studies restrict their attention to data where all patients have a history of mental illness. Our long-term goal is to build models that can compute individualized risk of suicide attempt/fatality for all ED patients; naturally, this includes those with and without a history of any particular illness. Hence our data set is much larger in size and scope than in prior studies.

In this work, we treat each ED/hospital visit in 2006-2009 by each patient as a separate observation. There are 772,923 such visits in the full data set. To each visit, we assign an outcome variable  $y^f$  of either 1 (presented in 2010 with a suicide attempt code, as defined above) or 0 (absence of suicide attempt code in 2010). For each visit, the raw predictors consist of the patient’s encrypted SSN, admission date, year, sex, age, race, insurance category, patient’s zip code and county of residence, etc. We do not list all predictors here for reasons of space—importantly, our data does not include textual annotations from medical professionals, survey responses, or numerical mental health evaluations. The vast majority of predictors are categorical; in these cases, we apply one-hot encoding.

The only visit variables that require special treatment are the diagnostic codes associated with the visit. Each visit is assigned one primary and up to six other Clinical Classifications Software (CCS) diagnostic codes. The CCS grouping system aggregates >14,000 ICD-9-CM diagnoses into 285 discrete, mutually exclusive, clinically meaningful category codes (e.g., “anxiety disorders”) that are useful for identifying patients in broad diagnosis groupings. In our analysis, for each visit, we aggregate all seven one-hot encodings of all CCS diagnostic codes. The result is a vector  $\mathbf{v} \in \mathbb{R}^{285}$  that tracks up to seven diagnoses received in one visit.

Within the 2006-2009 data corpus, patients may visit one or more times. To track a patient’s history, we include for a visit at time  $t$  the cumulative sum of all diagnosis vectors  $\mathbf{v}$  (computed as above) recorded at times less than or equal to  $t$ . We also record in a new column how many times the patient has been observed in the data set up to and including time  $t$ . By virtue of this cumulative encoding, each patient’s visit is transformed into a vector of fixed length. The end result is a full data matrix  $X^f$  of size  $772923 \times 494$ .

We randomly reshuffle the rows of  $(X^f, y^f)$  and split the result into a pretraining set  $(X, y)$  (80% of the rows) and a test set  $(X^t, y^t)$  (the remaining 20%). We record the column means and variances of  $X$ , discard columns with zero variance (i.e., unobserved diagnostic codes), and normalize all columns of  $X$  to have zero mean and unit variance. Note that we are left with  $p = 382$  columns.

Since only  $9804/618338 \approx 1.58\%$  of the  $y$  values in the pretraining set are equal to 1, we have an imbalanced classification problem. We sample with replacement from the rows of  $X$  associated with the minority ( $y = 1$ ) class to create a bootstrapped data set with balanced classes. The resulting data set  $(X^b, y^b)$  has  $N = 1217068$  rows.

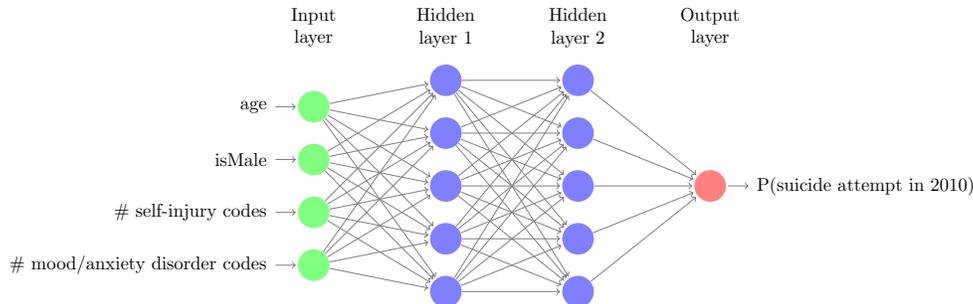


Figure 1: While our actual networks differ in the number of inputs, the number of units per layer, and the overall number of layers, the general architecture is as above.

### 3 Methods

As described above, deep neural networks have been underutilized in suicide prediction research. Hence we focus on neural network classifiers consisting of feedforward networks (equivalently, multilayer perceptrons) with dense, all-to-all connections between layers.

To illustrate the architectures of our networks, we present Figure 1. In this diagram, information flows from left to right, from the green input nodes to the red output nodes. Hidden layer  $i$  receives a vector  $\mathbf{h}^{(i-1)} \in \mathbb{R}^{n_{i-1}}$  of inputs from the layer to its left. Suppose there are  $n_i$  units or nodes in hidden layer  $i$ ; then the trainable parameters for this layer consist of  $\mathbf{W}_i$ , an  $n_{i-1} \times n_i$  weight matrix and  $\mathbf{b}_i \in \mathbb{R}^{n_i}$ , a bias vector. With these parameters, hidden layer  $i$  outputs  $\mathbf{h}^{(i)} = \phi(\mathbf{W}_i^T \mathbf{h}^{(i-1)} + \mathbf{b}_i)$  where  $^T$  denotes transpose and the activation function  $\phi: \mathbb{R} \rightarrow \mathbb{R}$  is applied element-wise. In our work, we use the scaled exponential linear unit (SELU)  $\phi(z) = \lambda(zI_{z>0}(z) + \alpha(e^z - 1)I_{z \leq 0}(z))$ , with constants  $\lambda \approx 1.0507$  and  $\alpha \approx 1.6733$  [12]. This activation function has been engineered to approximately preserve zero-mean, unit-variance normalization across many layers while avoiding vanishing/exploding gradients [12]. The use of the SELU activation renders batch normalization unnecessary [9].

With the convention that  $\mathbf{h}^{(0)} = \mathbf{x}$  where  $\mathbf{x} \in \mathbb{R}^p$  is the input vector, we have completely described propagation up to the final hidden layer. Suppose there are  $d$  total hidden layers, so that the output of the final hidden layer is  $\mathbf{h}^{(d)}$ . Then the output layer computes  $\sigma(\mathbf{w}_o^T \mathbf{h}^{(d)} + b_o)$  with trainable weight vector  $\mathbf{w}_o \in \mathbb{R}^{n_d}$  and scalar bias  $b_o \in \mathbb{R}$ . Here  $\sigma(z) = (1 + \exp(-z))^{-1}$  is the sigmoidal function; its output is the model’s probability of a suicide attempt in 2010, i.e.,  $P(y = 1|x, \theta)$ .

Hence our neural network computes the probability  $P(y = 1|x, \theta)$  where  $\theta$  stands for all trainable parameters (weights and biases). To train the model, we find  $\theta$  that minimizes the log likelihood on the training data,  $\ell(\theta) = \sum_i y_i \log P(y = 1|x_i, \theta) + (1 - y_i) \log[1 - P(y_i = 1|x_i, \theta)]$ . To minimize  $\ell$ , we employ either stochastic gradient descent (SGD) or SGD with momentum [24]. Gradients are computed using backpropagation. We leave further details to the code, available on GitHub (anonymous URL).

The only regularization technique we use is a variant of early stopping [9]. From the bootstrapped data set, we make a further split into the training set (80%) and a validation set (20%). We monitor the accuracy, sensitivity, and specificity on both the training and validation sets, and halt optimization when we detect insufficient improvement in our metrics on the validation set. Importantly, the test set is not used in this procedure.

All computations are carried out in Python using TensorFlow [1]. We use a rack-mounted server with two 8-core Intel Xeon E5-2620v4 (2.1GHz) CPUs, two NVidia Tesla P100 GPUs, and 128 GB of ECC RAM.

Method	Sens.	Spec.	Prec.	AUC	Method	Sens.	Spec.	Prec.	AUC
NN2	0.401	0.952	0.121	0.796	NN2, 662	0.710	0.889	0.527	0.896
NN4	0.390	0.956	0.127	0.802	NN4, 662	0.746	0.839	0.446	0.881
NN8	0.485	0.884	0.063	0.778	NN8, 662	0.779	0.597	0.251	0.776
NN2, $v \geq 2$	0.496	0.965	0.208	0.868	NN2, 651/7	0.610	0.923	0.390	0.882
NN4, $v \geq 2$	0.489	0.966	0.211	0.871	NN4, 651/7	0.616	0.910	0.356	0.873
NN8, $v \geq 2$	0.557	0.903	0.094	0.834	NN8, 651/7	0.703	0.742	0.179	0.806
NN2, $v \geq 3$	0.578	0.972	0.302	0.911	NN2, 659	0.780	0.910	0.627	0.929
NN4, $v \geq 3$	0.578	0.973	0.306	0.914	NN4, 659	0.752	0.894	0.581	0.910
NN8, $v \geq 3$	0.609	0.916	0.130	0.870	NN8, 659	0.819	0.651	0.313	0.832
NN2, $v \geq 4$	0.642	0.977	0.398	0.939	NN2, 660/1	0.530	0.945	0.370	0.854
NN4, $v \geq 4$	0.647	0.977	0.401	0.941	NN4, 660/1	0.541	0.940	0.357	0.862
NN8, $v \geq 4$	0.654	0.927	0.172	0.897	NN8, 660/1	0.618	0.797	0.156	0.796
NN2, $v \geq 5$	0.697	0.982	0.503	0.956	[22]-ML	0.37		0.27	
NN4, $v \geq 5$	0.703	0.980	0.489	0.958	[23]	0.96		0.75	0.83
NN8, $v \geq 5$	0.697	0.936	0.220	0.919	[22]-Clin.	0.081		0.129	

Table 1: Results on the test set, for which no bootstrapping has been applied; the prevalence of suicide attempts  $\approx 1.5\%$ . NN2/4/8 is our neural network model with, respectively, 2, 4, or 8 hidden layers; see the main text for details. Here  $v \geq n$  means that we only issue predictions for patients who have made at least  $n$  visits. CCS designations (e.g., 662) indicate that we only issue predictions for patients whose prior diagnoses includes the corresponding diagnostic codes; this is detailed in the main text. In the lower-right corner, we quote results from the literature. ML stands for the best machine learning model from [22], while Clin. denotes the accuracy of a human clinician. For all metrics, we define the “positive” class to be  $y = 1$ . Sens. = sensitivity, Spec. = specificity, Prec. = precision, AUC = area under the ROC curve.

## 4 Results and Discussion

Table 1 shows results for three trained networks: NN2, with  $d = 2$  hidden layers and 50 units per layer; NN4, with  $d = 4$  hidden layers and 50 units per layer; and NN8, with  $d = 8$  hidden layers and 20 units in each hidden layer except for the first which has 50 units. We chose the NN8 architecture so that the total number of weights is close to the total number of weights in the NN4 model.

All results described in this section are test set results. That is, we use the trained network to generate predictions for test set predictors  $X^t$ , after normalizing and discarding columns as described above.

In the upper-left corner of Table 1, we see the results for the entire test set. NN4 has the best results in categories other than sensitivity; in this metric, the NN8 model is the clear winner. Interestingly, the NN8 network is much more willing than NN2 or NN4 to issue a prediction of  $y = 1$ , i.e., that a given visit is associated with a future suicide attempt. With NN2 or NN4, bad choices of hyperparameters lead to models that predict  $y = 0$  100% of the time. In the remainder of the left half of Table 1, we have recorded the performance of the model when we set a threshold of  $n$  minimum visits in a patient’s record; in these  $v \geq n$  results, we simply do not issue a prediction if the patient has made less than  $n$  visits including the present one. As one would expect, the performance of all models improves as the amount of data we have about a given patient increases. The AUC values for the  $n \geq 5$  models are highly encouraging.

In the right-half of Table 1, we have broken down the test set results by the presence of prior CCS diagnostic codes: 662 corresponds to a prior diagnosis of self-injury, 651/657 to mood and anxiety disorders, 659 to schizophrenia and psychotic disorders, and 660/661 to alcohol- and substance-related disorders. Here we find several models with much higher sensitivity and precision (or positive predictive value) than on the test set as a whole. This is to be expected; past history of these particular diagnostic codes is known to be associated with future suicide attempts. In the lower-right corner of Table 1, we have quoted results from the literature. While the NN models outperform both machine and human predictions from Tran et al. [22], they do not achieve the sensitivity or precision of Walsh et al. [23]; this latter study restricts its modeling efforts to 5167 patients with a history of self-injury.

Our results motivate at least four areas for future work. First, we have not unpacked the features learned by the network, i.e., the mapping from the inputs  $\mathbf{x}$  to the last hidden layer  $\mathbf{h}^{(d)}$ . Understanding the features may help us develop more interpretable risk factors that help predict suicide attempts. Second, we have merely scratched the surface of neural network architectures, leaving unexplored recent developments designed to capture the temporal nature of EHRs [13, 4]. Third, we have not explored regularization techniques such as dropout [21], which may well improve generalization especially for deeper networks. Fourth, it may be advantageous to specialize our modeling efforts to patients with particular prior histories, as in [23]. We look forward to addressing these issues.

## References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. URL <https://arxiv.org/abs/1603.04467>.
- [2] Centers for Disease Control and Prevention. Web-based injury statistics query and reporting system (wisqars). *natl cent inj prev control centers dis control prev.*, 2013. URL <https://www.cdc.gov/injury/wisqars/fatal.html>.
- [3] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Interpretable deep models for ICU outcome prediction. In *AMIA Annual Symposium Proceedings*, volume 2016, page 371. American Medical Informatics Association, 2016. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5333206/>.
- [4] Zhengping Che, Yu Cheng, Zhaonan Sun, and Yan Liu. Exploiting Convolutional Neural Network for Risk Prediction with Medical Feature Embedding. *arXiv:1701.07474 [cs, stat]*, January 2017. URL <http://arxiv.org/abs/1701.07474>. arXiv: 1701.07474.
- [5] A. E. Crosby, L. Ortega, and C. Melanson. *Self-Directed Violence Surveillance: Uniform Definitions and Recommended Data Elements, Version 1.0*. Atlanta, GA: Centers for Disease Control and Prevention, 2011.
- [6] David Delgado-Gomez, Hilario Blasco-Fontecilla, AnaLucia A. Alegria, Teresa Legido-Gil, Antonio Artes-Rodriguez, and Enrique Baca-Garcia. Improving the accuracy of suicide attempter classification. *Artificial Intelligence in Medicine*, 52(3):165–168, July 2011. ISSN 09333657. doi: 10.1016/j.artmed.2011.05.004. URL <http://linkinghub.elsevier.com/retrieve/pii/S0933365711000595>.
- [7] David Delgado-Gomez, Hilario Blasco-Fontecilla, Federico Sukno, Maria Socorro Ramos-Plasencia, and Enrique Baca-Garcia. Suicide attempters classification: Toward predictive models of suicidal behavior. *Neurocomputing*, 92:3–8, September 2012. ISSN 09252312. doi: 10.1016/j.neucom.2011.08.033. URL <http://linkinghub.elsevier.com/retrieve/pii/S0925231212000835>.
- [8] Joseph C. Franklin, Jessica D. Ribeiro, Kathryn R. Fox, Kate H. Bentley, Evan M. Kleiman, Xieying Huang, Katherine M. Musacchio, Adam C. Jaroszewski, Bernard P. Chang, and Matthew K. Nock. Risk factors for suicidal thoughts and behaviors: A meta-analysis of 50 years of research. *Psychological Bulletin*, 143(2):187–232, 2017. ISSN 1939-1455, 0033-2909. doi: 10.1037/bul0000084. URL <http://doi.apa.org/getdoi.cfm?doi=10.1037/bul0000084>.
- [9] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning Series. MIT Press, 2016. ISBN 978-0-262-03561-3.
- [10] Krystl Haerian, Hojjat Salmasian, and Carol Friedman. Methods for identifying suicide or suicidal ideation in EHRs. In *AMIA Annual Symposium Proceedings*, volume 2012, page 1244. American Medical Informatics Association, 2012. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540459/>.
- [11] Theodoros Iliou, Georgia Konstantopoulou, Mandani Ntekouli, Dimitrios Lymberopoulos, Konstantinos Assimakopoulos, Dimitrios Galiatsatos, and George Anastassopoulos. Machine Learning Preprocessing Method for Suicide Prediction. In Lazaros Iliadis and Ilias Maglogiannis, editors, *Artificial Intelligence Applications and Innovations*, volume 475, pages 53–60. Springer International Publishing, Cham, 2016. ISBN 978-3-319-44943-2 978-3-319-44944-9. URL [http://link.springer.com/10.1007/978-3-319-44944-9\\_5](http://link.springer.com/10.1007/978-3-319-44944-9_5). DOI: 10.1007/978-3-319-44944-9\_5.
- [12] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems (NIPS)*, June 2017. URL <http://arxiv.org/abs/1706.02515>. arXiv: 1706.02515.
- [13] Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured Inference Networks for Nonlinear State Space Models. In *Thirty-First AAAI Conference on Artificial Intelligence*, February 2017. URL <https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14215>.
- [14] Medicode. *ICD-9-CM: International classification of diseases*. 9th revision, clinical modification edition, 1996.
- [15] Thuong Nguyen, Truyen Tran, Shivapratap Gopakumar, Dinh Phung, and Svetha Venkatesh. An evaluation of randomized machine learning methods for redundant data: Predicting short and medium-term suicide risk from administrative records and risk assessments. *arXiv preprint arXiv:1605.01116*, 2016. URL <https://arxiv.org/abs/1605.01116>.

- [16] Office of Statewide Healthcare Planning and Development. *MIRCal Edit Flag Description Guide: Emergency Department and Ambulatory Surgery Data*, volume 95811. Sacramento, CA, 2017.
- [17] Office of Statewide Healthcare Planning and Development. *MIRCal Edit Flag Description Guide: Inpatient Data*. Sacramento, CA, 2017.
- [18] Zachary E. Pittsenbarger and Rebekah Mannix. Trends in Pediatric Visits to the Emergency Department for Psychiatric Illnesses. *Academic Emergency Medicine*, 21(1):25–30, January 2014. ISSN 1553-2712. doi: 10.1111/acem.12282. URL <http://onlinelibrary.wiley.com/doi/10.1111/acem.12282/abstract>.
- [19] Daniele Ravi, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang. Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, 21(1):4–21, 2017.
- [20] Francisco Ruiz, Isabel Valera, Carlos Blanco, and Fernando Perez-Cruz. Bayesian non-parametric modeling of suicide attempts. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1853–1861, 2012. URL <http://papers.nips.cc/paper/4826-bayesian-nonparametric-modeling-of-suicide-attempts>.
- [21] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. URL <http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>.
- [22] Truyen Tran, Dinh Phung, Wei Luo, Richard Harvey, Michael Berk, and Svetha Venkatesh. An integrated framework for suicide risk prediction. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1410–1418. ACM, 2013. URL <http://dl.acm.org/citation.cfm?id=2488196>.
- [23] Colin G. Walsh, Jessica D. Ribeiro, and Joseph C. Franklin. Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, 5(3):457–469, 2017.
- [24] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv:1705.08292 [cs, stat]*, May 2017. URL <http://arxiv.org/abs/1705.08292>. arXiv: 1705.08292.
- [25] Nawal A. Zaher and Christopher D. Buckingham. Moderating the influence of current intention to improve suicide risk prediction. In *AMIA Annual Symposium Proceedings*, volume 2016, page 1274. American Medical Informatics Association, 2016. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5333240/>.