**Deep Learning Neural Network Derivation and Testing to Distinguish Acute Poisonings**

**Abstract**

**Background:** Acute poisoning is a significant global health burden, and the causative agent is often unclear. The primary aim of this pilot study was to develop a deep learning algorithm that predicts the most probable agent a poisoned patient was exposed to from a pre-specified list of drugs.

**Research design & methods:** Data were queried from the National Poison Data System (NPDS) from 2014 through 2018 for eight single-agent poisonings (acetaminophen, diphenhydramine, aspirin, calcium channel blockers, sulfonylureas, benzodiazepines, bupropion, and lithium). Two Deep Neural Networks (PyTorch and Keras) designed for multi-class classification tasks were applied.

**Results:** There were 201,031 single-agent poisonings included in the analysis. For distinguishing among selected poisonings, PyTorch had specificity of 96%, accuracy of 75%, precision of 70%, recall of 75%, and a F1-score of 71%. Keras had specificity of 96%, accuracy of 74%, precision of 67%, recall of 72%, and a F1-score of 69%. The best performance was achieved in the diagnosis of single-agent poisoning by sulfonylureas, acetaminophen, benzodiazepines, and aspirin, in PyTorch (F1-score = 92%, 87%, 83%, and 70%, respectively) and Keras (F1-score = 91%, 84%, 83%, and 66%, respectively).

**Conclusion:** Deep neural networks can potentially aid in distinguishing the causative agent of acute poisoning. This study used a small list of drugs, with polysubstance ingestions excluded. The clinical utility is untested, and with further development, this may become a diagnostic aid for clinicians without a background in toxicology who care for poisoned patients with unreliable histories.

**Keywords:** Deep learning, Machine learning, Poisoning,Toxicity, PyTorch, Keras

**1. Introduction**

Acute poisoning is a significant global health burden, with regional poison control centers in the United States reporting approximately 2.1 million human poison exposure cases annually [[1](#_ENREF_1" \o "Gummin, 2020 #1)]. Obtaining a reliable exposure history may be difficult in poisoned patients due to the severity of poisoning, obfuscation by the patient, and variable or overlapping clinical features. Mental health conditions, such as suicidality or drug intoxication, may also complicate obtaining an accurate toxicologic exposure history. Clinician decision-making, in general, is guided by the history of the presenting illness, contextualized with a patient’s past medical history and clinical signs and symptoms. For cases involving overdose or poisoning, clinical features of the presentation can suggest the causative agent [[2](#_ENREF_2" \o "Erickson, 2007 #2),[3](#_ENREF_3" \o "Thanacoody, 2020 #3)].

Artificial Intelligence (AI) is focused on mimicking human decision-making based on logic and decision trees. Machine Learning (ML) refers to a subset of AI using [statistical models](https://en.wikipedia.org/wiki/Statistical_model) to perform tasks based on inference rather than requiring explicit instructions and allowing the algorithm to improve with experience. ML attempts to learn data patterns by using tailored algorithms. While high-quality input data are critically important for training and validation, a large dataset improves performance and can overcome some weaknesses linked to smaller datasets [[4](#_ENREF_4)]. ML algorithms use ‘[training data](https://en.wikipedia.org/wiki/Training_data)’ to derive a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model), then apply the model to a set of ‘testing data’ to arrive at predictions. The types of learning used by ML algorithms can be divided into supervised and unsupervised learning based on the availability of labeled data [[4](#_ENREF_4)]. Deep Learning is a subset of ML and uses a Deep Neural Network (DNN) with layered calculations in an attempt to mimic the human brain in data processing. DNNs are composed of three main layers: the input layer receives input, the hidden layer (or layers) performs calculations on data from the input layer, and the output layer outputs information from the hidden layer. Deep Learning particularly excels where a large data volume is best suited for classification tasks similar to our study [[5](#_ENREF_5)].

While clinical applications of ML are still nascent, ML has already been used in varied applications including patient risk stratification, Alzheimer’s disease detection, echocardiogram analysis, interpretation of physiologic data, and many other exploratory applications [[6-10](#_ENREF_6" \o "Chang, 2019 #6)]. In medical toxicology, few studies have used DNNs on national poisoning data to identify the cause of poisoning [[11-15](#_ENREF_11" \o "Chary, 2021 #11)]. The primary aim of this pilot study was to develop a DNN that predicts the most probable agent a poisoned patient has been exposed to among eight possible drugs based on clinical features (i.e., signs and symptoms). Due to the complexity of polysubstance exposures, this study focused on single-agent exposures.

**2. Methods**

The National Poison Data System (NPDS) collects call data from all regional poison centers in the United States. This study was formally exempted by the Colorado Multiple Institutional Review Board (COMIRB#: 22-1088). De-identified NPDS data were queried from 2014 through 2018 for exposures in patients aged 0-89 years for eight single-agent poisonings (acetaminophen, diphenhydramine, aspirin, calcium channel blockers, sulfonylureas, benzodiazepines, bupropion, and lithium). Authors with expertise in medical toxicology selected these drugs and drug classes based on the complexity of their clinical presentations and overlap in clinical effects. Within the drug classes of benzodiazepines, calcium channel blockers, and sulfonylureas, specific agents were assumed for this study to cause similar clinical manifestations. This study did not require review by the Colorado Multiple Institutional Review Board as it is not considered human subject research according to 45 CFR Part 46.102.

The acquired dataset from NPDS has 133 features. In ML, a feature of the data is defined as an independent variable and typically used as an input to predict a dependent variable. The dataset for this study has 201,031 entries, each corresponding to one single-agent poisoning case. For each case, the first feature is age, the second is gender, and the rest of the 131 features are symptoms included in NPDS [[16](#_ENREF_16" \o ",  #16)]. All included features are binary except age which is a continuous variable. For this algorithm, the dependent (target) variable was “Product” or drug. Drugs in this study were in 1 of 8 classes, and we use the terms class product and drug interchangeably.

For binary features, the ‘Feature Coverage’ (FC) index measures the percentage of data entries that have feature F (F = 1) in which the poisoning agent has been drug D. Therefore, for a drug A and feature F, a high FC means that if a poisoning case has feature F, it is highly likely the reason for the poisoning has been drug A. The formula to calculate this percentage is:

FC of feature F for drug A,

Wherein,

is the number of samples where feature F is present (value of feature F is 1)

is the number of samples with feature F present among drug A samples

The FC for each binary feature and 8 drugs are shown in Supplementary Figure 1. Several features showed higher FC for a particular drug, but if the feature appears in only a few samples, it is not useful to predict the drug. For example, in Supplementary Figure 1 ‘Deafness’ has high FC for aspirin, which suggests that poisoning cases with deafness are likely to be from aspirin. However, the number of cases with deafness is too low relative to the number of cases with aspirin. A separate parameter, Class Coverage (CC), can account for this limitation in FC. For each drug A and binary feature F, CC measures the percentage of data entries in which the poisoning agent is drug A and which have the feature F (F = 1) over the total number of cases in which the poisoning agent is drug A. If a feature has a high CC for drug A, then a case exposed to drug A has a high probability of the presence of feature F.

CC of feature F for drug A,

Wherein,

is the number of drugs A samples

is the number of samples where feature F is present among drug A samples

Supplementary Figure 1 includes CC for drug and binary features. If a feature has both high FC and CC for a drug, the feature is important to predict that drug.

Since the dependent (target) variable has 8 categorical values, this is a classification problem. Since the causative poisoning agent in this dataset is known, this problem can also be classified as a supervised learning problem. The first step of ML is preparing the data in a format accepted by the model, termed data pre-processing. In this study, pre-processing included missing-value resolution and data normalization. Cases with a missing feature or target value (total 8,242) were excluded, and 201,031 cases were including after missing-value resolution. For data normalization, since all features are binary except “Age”, we normalized “Age” based on its mean and standard deviation to be the same order of magnitude as the Boolean variables, leading to less model variance.

Before model training occurs, the model is configured based on the number of hidden layers, size of each layer, learning rate, etc. These are pre-specified and termed hyper-parameters of the model. Once the data is preprocessed, the final step is to randomly divide it into three separate sets: training, validation, and testing. To determine optimal hyper-parameters, typically different models are trained with the training set, then performance is measured on the validation set. The model with the best performance on the validation set is considered the best model, and performance is then evaluated on the testing dataset to provide an unbiased estimate of model performance.

Finally, using random search methods, a series of experiments were conducted with variations of the DNN to select optimal hyper-parameters including the number of layers, number of units (i.e., neurons) in each layer, and the dropout rate. An intuitive trial-and-error approach was additionally used to optimize the following hyper-parameters: learning rate, epoch (number of training rounds), and batch size. The testing dataset was not used for hyper-parameter tuning.

We developed two fully connected models based on PyTorch and Keras frameworks [[17-19](#_ENREF_17" \o "Chen, 2016 #17)] with Tensorflow [[20](#_ENREF_20" \o "Abadi, 2016 #20)] backend. PyTorch is widely used in DNN models given its ease in debugging and computational efficiency. The performance matrices used are overall accuracy, specificity, precision, recall, and F1-score for the multi-class classification algorithm. F1-score is the geometric mean of precision and recall, combining their scores. Precision (positive predictive value) indicates the percentage of predictions that were correct. Recall (sensitivity) indicates the percentage of drug products predicted correctly by the model. Accuracy is the percentage of correct predictions across all included drug products.

Confusion matrices were summarized for training and testing cohorts in each DNN model. A confusion matrix summarizes performance of a classification algorithm, including information on the algorithm’s correct assumptions and errors, the total test cases available for each drug product, and the number classified correctly. The confusion matrix also indicates whether some detections were false, and how many detections were assigned to each product. For each drug class, a confusion matrix displays true positives, false positives, true negatives, and false negatives. In this context, positive means being classified as a particular drug class.

Classification accuracy alone can be misleading if there are unequal observations in each class or more than two classes in a dataset. To confirm that training, validation, and testing datasets contain similar feature characteristics, the feature distribution of Age (only numeric feature) in Figure 1 and the percentage for all binary features are listed for training, validation, and testing sets (Supplementary Figure 2). It can be visually verified from the plots that the three datasets have similar features.

**2.1. Experimental Configurations**

This problem can be considered an 8-class classification problem based on 133 input features. In deep learning, an n-class classification problem is typically modeled by a DNN that uses features as the input layer, performs calculations in hidden layers, and outputs n probabilities using an n-class SoftMax layer. SoftMax is a function that takes in a vector of n numbers and outputs a probability mass function with n events associated with the input values. The larger each input is, the higher the output probability is, so once the model outputs 8 probabilities, the node with the highest probability is assigned as the drug class for that sample.

After hyper-parameter tuning on validation data, our final architecture was selected as a DNN with an input layer, 4 fully connected hidden (intermediate) layers, and one output layer. The intermediate layers contained 512, 1024, 1024, and 256 neurons, respectively. The output layer contained 8 neurons with a SoftMax activation function which indicates the probability of poisoning being caused by each drug class. The ReLU (Rectified Linear Unit) activation function was applied at each layer output, and the dropout layer was used with a dropping probability of 0.15 at each hidden layer. ReLU activation functions improve training stability, and dropout layers minimize over-fitting the model to the training dataset. A detailed model description is presented in Figure 2.

We developed two models using two popular Deep Learning libraries (PyTorch and Keras with TensorFlow backend) with the architecture described above. The categorical cross-entropy loss function was used as the optimization criterion for training both models and updating weights [[21](#_ENREF_21)].

Model training occurred using the Adam optimizer with a default set of parameters described by Kinga and Adam (2014) and trained for 200 epochs. Adam is an upgrade to the gradient descent algorithm which uses the concept of momentum to approach the minima in a more stable manner. Both models were trained with the same graphics processing unit and central processing unit configuration [[22](#_ENREF_22)].

**3. Results**

There were 201,031 single-agent poisonings included in the DNNs. Frequencies of agents involved in the poisonings are displayed in Table 1. Figures 3 and 4 show performance of two DNN models (PyTorch and Keras) on training and the testing data. The PyTorch DNN had an overall accuracy of 75.7%, 75.1%, and 75.5%, and an F1-score of 72.4%, 71.3%, and 71.9% for training, validation, and testing groups, respectively, in distinguishing between selected drugs (Figure 3). The Keras DNN had an overall accuracy of 80.9%, 74.2%, and 74.6%, and an F1-score of 79.8%, 69.3%, and 69.6% for training, validation, and testing groups, respectively, in distinguishing between selected drugs (Figure 4).

Table 2 illustrates performance of the two DNN models on four best-classifying drugs: sulfonylureas, acetaminophen, benzodiazepines, and aspirin. The PyTorch and Keras DNNs both had the best performance in diagnosing sulfonylureas exposure, followed by acetaminophen, benzodiazepines, then aspirin. Figures 3 and 4 show performance of the two models on the eight drug products. Score matrices show specificity, precision, recall, mean (F1-score), and accuracy for each product. For example, for sulfonylureas (Product 7) in the testing set score matrix (Figure 3), precision of 0.906 means 90.6% of testing cases predicted as due to sulfonylureas are correct. Accuracy of 0.755 indicates 75.5% of cases were classified by the model correctly among all testing cases.

Table 3 displays mean F1-scores and overall accuracy for drug products for the two DNN models. Both models had the highest accuracy in diagnosing poisoning by sulfonylureas, then acetaminophen, benzodiazepines, then aspirin. The highest F1-score was achieved for discriminating sulfonylurea poisoning (92%).

**4. Discussion**

This study applied two DNN models to predict the most probable drug poisoned patients were exposed to among eight single-agent poisonings. The results of this pilot study with a limited number of selected drug possibilities suggest a DNN can identify the causative drug in poisoning with an overall accuracy of 75% and precision of 70%. This pilot study achieved the best performance in single-agent poisoning classification across both DNN models for sulfonylureas, acetaminophen, benzodiazepines, aspirin, and diphenhydramine, respectively.

The predictive capability of our classifier is mainly dictated by data variance, Class Coverage, and Feature Coverage for the presence of a symptom with each drug agent. Increased overlap in symptoms complicates the classifier in determining a decision boundary to separate drug classes [[23](#_ENREF_23" \o "Gu, 2017 #23),[24](#_ENREF_24" \o "Xiong, 2013 #24)]. Although sulfonylurea drugs constitute only 1.34% of the 201,031-case dataset, the positive and negative classes (based on presence or absence of a symptom) possess distinct variance statistics per feature (Supplementary Figure 1).

The overall model accuracy and performance indicates the models can predict the included drug products and classify cases based on clinical features. Moreover, there was correlation between FC and CC values (Supplementary Figure 1) and model performance (Figures 3c and 4c). Results from this study are largely consistent with widely accepted clinical manifestations of drug toxicity from the included drug products. For example, the models identified benzodiazepine toxicity based on drowsiness and lethargy (Supplementary Figure 1). The clinical hallmark of sulfonylurea toxicity is hypoglycemia, which resulted in both high CC and FC for sulfonylureas. These instances of high CC and FC are one reason that the model can distinguish sulfonylureas successfully. The two models accurately identified sulfonylurea toxicity, but of note, sulfonylureas were the only included drug class expected to frequently cause hypoglycemia. It is common for ML and Deep Learning applications in medicine to focus on prognosis, diagnosis, or differentiation of clinical groups (e.g., a group with pathology) [[25](#_ENREF_25" \o "Triantafyllidis, 2019 #25)]. There is variability in how clinical success is defined for DNN or ML algorithms in medicine based on the evaluation matrix emphasized and the cutoff used to consider clinical success and utility. There is no best-practice cutoff value, and clinicians may determine what performance threshold is acceptable in their practice [[26-29](#_ENREF_26" \o "Arisholm, 2010 #26)].

Applications of ML in medicine and medical toxicology research is accelerating. A study by Nogee et al. (2020) applied ML based multi-class classification on clinical features to determine the causative agent in selected poisonings with a mean overall accuracy of 61.9% (Decision Trees model 51.4%, XGBoost model 65.9%) [[14](#_ENREF_14" \o "Nogee, 2020 #14)]. Different multi-class classification algorithms were used (e.g., Bayesian Naïve Bayes, DT, Support Vector Machines, Random Forests of Trees, and XGB) to predict exposure to toxicants (anticholinergics, acetaminophen, bupropion, benzodiazepines, carbon monoxide, ethanol, opioids, clonidine, selective serotonin reuptake inhibitors, and sympathomimetics). The models by Nogee et al. performed best in classifying carbon monoxide, opioid, and benzodiazepine exposures [[14](#_ENREF_14" \o "Nogee, 2020 #14)]. Their models had lower performance in predicting bupropion, similar to our findings. Additionally, they examined a lower number of cases (n= 2057 cases) in a single-center design, and their models had lower accuracy than the models in the current study. Although there was high accuracy, specificity, F1-scores, and precision for most drugs in our study, recall for bupropion was relatively low. Model performance was better for some poisoning exposures than others, which may have been due to more data in some classifications than others. Another potential explanation for varied recall between classes could be incomplete data or data inaccuracy in some classes as seen in the bupropion drug class.

ML algorithms in toxicology have been studied beyond the identification of clinical syndromes. A study by Dong et al. (2019) examined opioid overdose prediction using the Random Forest method, achieving recall of 85.7% and accuracy of 99.2% [[30](#_ENREF_30)]. Zhu et al. (2018) and Xu et al. (2019) evaluated ML analysis of blood tests to identify glyphosate [[31](#_ENREF_31)] and phenanthrene [[32](#_ENREF_32)] poisoning in rats. ML has been explored to predict the prognosis of paraquat-poisoned patients using biochemical indexes [[32-34](#_ENREF_32)], seizures from tramadol poisoning [[35](#_ENREF_35" \o "Behnoush, 2021 #35)], older adults at risk of adverse drug events [[36](#_ENREF_36" \o "Ouchi, 2018 #36)], children at risk of lead poisoning [[37](#_ENREF_37" \o "Potash, 2020 #37)], and ototoxicity of cigarettes/pesticides [[38](#_ENREF_38" \o "Tomiazzi, 2019 #38)]. ML has also been applied to identify poisonous mushrooms via mobile phone, called the Mushroom Diagnosis Assistance System (MDAS) [[39](#_ENREF_39)].

Limitations in the present study include that its retrospective design may have led to a bias and the inability to control confounding factors. Incomplete or inaccurate documentation at the time of the poison center call, either from lack of information conveyed to the poison specialist or from a transcription error, can result in inaccurate data capture [[40](#_ENREF_40" \o "Hoffman, 2007 #40)]. Diagnostic test results, which may aid in diagnosing a poisoning agent when clinical features are inadequate, were not used in this study. In this pilot study (and thus somewhat preliminary), we included just eight common single-agent poisonings (acetaminophen, diphenhydramine, aspirin, calcium channel blockers, sulfonylureas, benzodiazepines, bupropion, and lithium). Also, we considered only single-agent exposures due to the complexity of poly-substance exposure, limiting its generalizability. Complex pharmacological interactions between substances can lead to unwanted effects and different presentations. Poly-substance exposure alters the medication's effect on the body and it may cause unexpected side effects, delay, decrease, or enhance the action of either drug. The poisonings other than these agents are impossible to predict using the proposed models. So, further research for developing these models considering more drug agents and multiple exposures is needed.

Future studies should expand the models to additional drugs and substances, incorporate diagnostic test results when relevant, and prospectively validate model performance. It is also important to compare the models with human level performance (HLP). This requires appropriate selection of human participants for accuracy and generalizability. Measuring HLP adds significant complexity to a study, and while critically important prior to real-world implementation, it was beyond the scope of this pilot study.

This study demonstrates that DNNs may aid in distinguishing the causative agent in cases of acute poisoning. The clinical utility is untested, and with further development, this may become a diagnostic aid for clinicians without a background in toxicology who care for poisoned patients with unreliable histories. With developments they may also become useful for clinicians with expertise in medical toxicology, but this would likely require higher model performance to match or outperform HLP among this group. DNN algorithms are not intended to supersede clinical judgment but have the potential to serve as a powerful clinical decision-making adjunct.

**5. Conclusion**

A large national retrospective dataset was used to develop two DNN models to predict the causative drug in poisoned patients. Deep neural networks can potentially aid in distinguishing the causative agent of acute single exposures. On the Test Set of data, the PyTorch and Keras DNN models distinguished among selected poisonings with an overall accuracy of 75.5% and 74.6%, and precision of 70% and 67.8%, respectively. The PyTorch and Keras DNNs both had the best performance in diagnosing sulfonylureas exposure, followed by acetaminophen, benzodiazepines, and then aspirin. The highest F1-score was achieved for discriminating sulfonylurea poisoning (92%). The clinical utility of this models is untested, and with further development, it may become a diagnostic aid for clinicians without a background in toxicology.

**Disclaimers**

The American Association of Poison Control Centers (AAPCC; http://www.aapcc.org) maintains the national database of information logged by the country’s poison centers (PCs). Case records in this database are from self-reported calls: they reflect only information provided when the public or healthcare professionals report an actual or potential exposure to a substance (e.g., ingestion, inhalation, topical exposure, etc.) or request information/educational materials. Exposures do not necessarily represent a poisoning or overdose. The AAPCC is not able to completely verify the accuracy of every report made to member centers. Additional exposures may go unreported to PCs and data referenced from the AAPCC should not be construed to represent the complete incidence of national exposures to any substance(s).

**Declarations:**

**Declaration of Interest:**

The authors have no relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript. This includes employment, consultancies, honoraria, stock ownership or options, expert testimony, grants or patents received or pending, or royalties.

This paper was not funded.

**Acknowledgments**

The authors would like to thank the American Association of Poison Control Centers’ National Poison Data system for providing data for this study.

**Author Contributions**

OM and CH designed the study. OM, CH, HD, AA, AB, JS, SN, MSN,FG contributed to writing the draft and revising the manuscript. All authors approved the final version of the manuscript.

**Data Availability**

The datasets which were analyzed during this study are available from the corresponding author upon any reasonable request with permission of the National Poison Data System (NPDS) administrator.

**References**

1. Gummin DD, Mowry JB, Beuhler MC, et al. 2019 Annual Report of the American Association of Poison Control Centers’ National Poison Data System (NPDS): 37th Annual Report. Clin Tox. 2020;58(12):1360-1541.

2. Erickson TB, Thompson TM, Lu JJ. The approach to the patient with an unknown overdose. Emerg Med Clin North Am. 2007;25(2):249-281.

3. Thanacoody R. Principles of assessment and diagnosis of the poisoned patient. Medicine. 2020;48(3):156-159.

4. Deo RC. Machine learning in medicine. Circulation. 2015;132(20):1920-1930.

5. El-Mahelawi JK, Abu-Daqah JU, Abu-Latifa RI, et al. Tumor Classification Using Artificial Neural Networks. International Journal of Academic Engineering Research (IJAER). 2020;4(11):8-15.

6. Chang Y-S, Park H, Hong SH, et al. Predicting cochlear dead regions in patients with hearing loss through a machine learning-based approach: A preliminary study. PloS one. 2019;14(6):e0217790.

7. Metsker OG, Yakovlev AN, Ilin A, et al. Echocardiography Population Study in Russian Federation for 4P Medicine Using Machine Learning. Stud Health Technol Inform. 2019;261:137-142.

8. Patterson BW, Engstrom CJ, Sah V, et al. Training and interpreting machine learning algorithms to evaluate fall risk after emergency department visits. Med Care. 2019;57(7):560.

9. Rush B, Celi LA, Stone DJ. Applying machine learning to continuously monitored physiological data. J Clin Monit Comput. 2019;33(5):887-893.

10. Uspenskaya-Cadoz O, Alamuri C, Wang L, et al. Machine Learning Algorithm Helps Identify Non-Diagnosed Prodromal Alzheimer’s Disease Patients in the General Population. J Prev Alzheimers Dis. 2019;6(3):185-191.

11. Chary M, Boyer EW, Burns MM. Diagnosis of Acute Poisoning Using Explainable Artificial Intelligence. Comput Biol Med. 2021;134:104469.

12. Chary M, Burnsa M, Boyerb E. Tak: the computational toxicological machine. 39th International Congress of the European Association of Poisons Centres and Clinical Toxicologists (EAPCCT) 21-24 May 2019, Naples, Italy. Clin Tox. 2019;57(6):482.

13. Chary MA, Manini AF, Boyer EW, et al. The role and promise of artificial intelligence in medical toxicology. J Med Toxicol 2020;16(4):458-64.

14. Nogee D, Haimovich A, Hart K, et al. Multiclass Classification Machine Learning Identification of Common Poisonings. North American Congress of Clinical Toxicology (NACCT) Abstracts 2020. clin tox. 2020;58(11):1083-1084.

15. Nogee D, Tomassoni A. Development of a prototype software tool to assist with toxidrome recognition. North American Congress of Clinical Toxicology (NACCT) Abstracts 2018. clin tox. 2018;56(10):1049-1049.

16. American Association of Poison Control Centers. National Poison Data System (NPDS): NPDS coding users’ manual© version 3.1. 2014. Available at: https:// aapcc.s3.amazonaws.com/pdfs/ member-resources/NPDS\_Coding\_ Users\_Manual\_v3.1\_07May2014.pdf. Accessed October 2, 2017.

17. Chen T, Guestrin C. XGBoost: a scalable tree boosting system In: Proceedings of the 22Nd ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining. New York, NY: ACM; 2016 Aug 13 (pp. 785-794). 2016.

18. Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA2017. p. 3146-3154.

19. Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst. 2019;32:8026-37.

20. Abadi M, Barham P, Chen J, et al., editors. Tensorflow: A system for large-scale machine learning. In12th USENIX symposium on operating systems design and implementation (OSDI 16) 2016 (pp. 265-283). 2016.

21. Zhang Z, Sabuncu M. Generalized cross entropy loss for training deep neural networks with noisy labels. 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), eprint arXiv:1805.078362018.

22. Kingma DP, Ba J. Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations, San Diego, 2015. arXiv preprint arXiv:1412.69802014.

23. Gu Y, Cheng L. Classification of class overlapping datasets by kernel-MTS method. Int J Innov Comput Inf Control. 2017;13(5):1759-1767.

24. Xiong H, Li M, Jiang T, et al. Classification algorithm based on NB for class overlapping problem. Appl Math. 2013;7(2L):409-415.

25. Triantafyllidis AK, Tsanas A. Applications of machine learning in real-life digital health interventions: review of the literature. J Med Internet Res. 2019;21(4):e12286.

26. Arisholm E, Briand LC, Johannessen EB. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. J Syst Softw. 2010;83(1):2-17.

27. Barletta L, Giusti A, Rottondi C, et al., editors. QoT estimation for unestablished lighpaths using machine learning. InOptical Fiber Communication Conference 2017 Mar 19 (pp. Th1J-1). Optical Society of America; 2017.

28. Cheng M-Y, Kusoemo D, Gosno RA. Text mining-based construction site accident classification using hybrid supervised machine learning. Autom Constr. 2020;118:103265.

29. Johannessen EB. Data mining techniques, candidate measures and evaluation methods for building practically useful fault-proneness prediction models. Master’s thesis, University of Oslo, Department of Informatics, Oslo, Norway. 2008.

30. Dong X, Rashidian S, Wang Y, et al., editors. Machine Learning Based Opioid Overdose Prediction Using Electronic Health Records. AMIA Annu Symp Proc. 2020 Mar 4;2019:389-398. 2019: American Medical Informatics Association.

31. Zhu J, Zhao X, Li H, et al. An effective machine learning approach for identifying the glyphosate poisoning status in rats using blood routine test. IEEE Access. 2018;6:15653-15662.

32. Xu Y, Yu K, Wang P, et al. A new hybrid machine learning approach for prediction of phenanthrene toxicity on mice. IEEE Access. 2019;7:138461-138472.

33. Chen H, Hu L, Li H, et al. An effective machine learning approach for prognosis of paraquat poisoning patients using blood routine indexes. Basic Clin Pharmacol Toxicol. 2017;120(1):86-96.

34. Wen C, Lin F, Huang B, et al. Metabolomics Analysis in Acute Paraquat Poisoning Patients Based on UPLC-Q-TOF-MS and Machine Learning Approach. Chem Res Toxicol. 2019;32(4):629-637.

35. Behnoush B, Bazmi E, Nazari S, et al. Machine learning algorithms to predict seizure due to acute tramadol poisoning. Hum Exp Toxicol. 2021;40(8):1225-33.

36. Ouchi K, Lindvall C, Chai PR, et al. Machine learning to predict, detect, and intervene older adults vulnerable for adverse drug events in the emergency department. J Med Toxicol. 2018;14(3):248-252.

37. Potash E, Ghani R, Walsh J, et al. Validation of a machine learning model to predict childhood lead poisoning. JAMA Netw Open. 2020;3(9):e2012734-e2012734.

38. Tomiazzi JS, Pereira DR, Judai MA, et al. Performance of machine-learning algorithms to pattern recognition and classification of hearing impairment in Brazilian farmers exposed to pesticide and/or cigarette smoke. Environ Sci Pollut Res. 2019;26(7):6481-6491.

39. Al-Mejibli IS, Abd DH. Mushroom Diagnosis Assistance System Based on Machine Learning by Using Mobile Devices. Journal of Al-Qadisiyah for computer science and mathematics. 2017;9(2):103-113.

40. Hoffman RS. Understanding the limitations of retrospective analyses of poison center data. Clin tox. 2007;45(8):943-945.

**Table 1:** Frequency of Drugs Involved in Included Single-agent Poisonings from the National Poison Data System, 2014-2018.

|  |  |  |
| --- | --- | --- |
|  | **Frequency** | **Percent (%)** |
| Acetaminophen | 38,033 | 18.92 |
| Aspirin | 15,268 | 7.59 |
| Benzodiazepines | 68,536 | 34.09 |
| Bupropion | 12,328 | 6.13 |
| Calcium channel blocker | 2,945 | 1.46 |
| Diphenhydramine | 51,798 | 25.76 |
| Lithium | 9,434 | 4.69 |
| Sulfonylureas | 2,689 | 1.34 |
| **Total** | 201,031 | 100.00 |

**Table 2:** Precision and Recall of the Two DNN Models on the Top-performing Drug Categories

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DNN Model/Type of Poisonings** | PyTorch | | | Keras | |
| Precision | | Recall | Precision | Recall |
| Sulfonylureas | 0.89 | 0.93 | | 0.95 | 0.91 |
| Acetaminophen | 0.85 | 0.89 | | 0.82 | 0.92 |
| Benzodiazepines | 0.86 | 0.79 | | 0.82 | 0.87 |
| Aspirin | 0.74 | 0.66 | | 0.83 | 0.62 |

DNN: Deep Neural Network

**Table 3:** F1-scores and Overall Accuracy for Test Data of the DNN Models

|  |  |  |
| --- | --- | --- |
| **Drug exposure** | PyTorch | Keras |
| Acetaminophen | 0.87 | 0.84 |
| Aspirin | 0.70 | 0.66 |
| Benzodiazepines | 0.83 | 0.83 |
| Bupropion | 0.49 | 0.45 |
| Calcium channel blockers | 0.54 | 0.53 |
| Diphenhydramine | 0.67 | 0.68 |
| Lithium | 0.70 | 0.63 |
| Sulfonylureas | 0.92 | 0.91 |
| **Mean F1-score** | **0.71** | **0.69** |
| **Overall accuracy** | **0.75** | **0.74** |

DNN: Deep Neural Network

**Figure 1:** Feature Distribution Plot for ‘Age’ in Training, Validation and Testing Data

**Chart

Description automatically generated**

***A high-resolution version of Figure 1 can be found as a supplement file with name “Fig-1.png.”***

A picture containing diagram

Description automatically generated**Figure 2:** Overall Architecture of the Proposed Model

**Figure 3A:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Training Group)

Chart

Description automatically generated

i) Performance Characteristics for PyTorch-based Model (on Training Set)

Graphical user interface, application

Description automatically generated

ii) Confusion Matrix of PyTorch-based Model (on Training Set)

**Figure 3B:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Validation Group)

Chart

Description automatically generated

i) Performance Characteristics for PyTorch-based Model (on Validation Set)

Graphical user interface, application

Description automatically generated ii) Confusion Matrix of PyTorch-based Model (on Validation Set)

**Figure 3C:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Test Group)

Chart

Description automatically generated

i) Performance Characteristics for PyTorch-based Model (on Test Set)

Graphical user interface, application

Description automatically generated

ii) Confusion Matrix of PyTorch-based Model (on Test Set)

**Figure 4A:** Neural Network Model Characteristic and Confusion Matrix using Keras (Train Group)

Chart

Description automatically generated

i) Performance Characteristics for Keras-based Model (on Training Set)

Graphical user interface, application

Description automatically generated

ii) Confusion Matrix of Keras-based Model (on Training Set)

**Figure 4B:** Neural Network Model Characteristic and Confusion Matrix using Keras (Validation Group)

Chart

Description automatically generated

i) Performance Characteristics for Keras-based Model (on Validation Set)

Graphical user interface, application

Description automatically generated

ii) Confusion Matrix of Keras-based Model (on Validation Set)

**Figure 4C:** Neural Network Model Characteristic and Confusion Matrix using Keras (Test Group)

Graphical user interface, chart

Description automatically generated

i) Performance Characteristics for Keras-based Model (on Test Set)

Graphical user interface, application

Description automatically generated

ii) Confusion Matrix of Keras-based Model (on Test Set)

**A picture containing text, outdoor

Description automatically generatedSupplementary Figure 1:** Feature Coverage and Class Coverage Percentage Plot

*A high-resolution version of the figure can be found as a supplement file “supp Fig-1.png.”*

**Supplementary Figure 2:** Distribution of Presence of Binary Features in Training, Validation and Testing Data

**A picture containing diagram

Description automatically generated**

*A high-resolution version of the figure can be found as a supplement file “supp Fig-2.png.”*

**Tables and Figures’ legends:**

**Table 1:** Frequency of Drugs Involved in Included Single-agent Poisonings from the National Poison Data System, 2014-2018

**Table 2:** Precision and Recall of the Two DNN Models on the Top-performing Drug Categories

**Table 3:** F1-scores and Overall Accuracy for Test Data of the DNN Models

**Figure 1:** Feature Distribution Plot for ‘Age’ in Training, Validation and Testing Data

***A high-resolution version of Figure 1 can be found as a supplement file with name “Fig-1.png.”***

**Figure 2:** Overall Architecture of the Proposed Model

**Figure 3A:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Training Group)

i) Performance Characteristics for PyTorch-based Model (on Training Set)

ii) Confusion Matrix of PyTorch-based Model (on Training Set)

**Figure 3B:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Validation Group)

i) Performance Characteristics for PyTorch-based Model (on Validation Set)

ii) Confusion Matrix of PyTorch-based Model (on Validation Set)

**Figure 3C:** Neural Network Model Characteristic and Confusion Matrix using PyTorch (Test Group)

i) Performance Characteristics for PyTorch-based Model (on Test Set)

ii) Confusion Matrix of PyTorch-based Model (on Test Set)

**Figure 4A:** Neural Network Model Characteristic and Confusion Matrix using Keras (Train Group)

i) Performance Characteristics for Keras-based Model (on Training Set)

ii) Confusion Matrix of Keras-based Model (on Training Set)

**Figure 4B:** Neural Network Model Characteristic and Confusion Matrix using Keras (Validation Group)

i) Performance Characteristics for Keras-based Model (on Validation Set)

ii) Confusion Matrix of Keras-based Model (on Validation Set)

**Figure 4C:** Neural Network Model Characteristic and Confusion Matrix using Keras (Test Group)

i) Performance Characteristics for Keras-based Model (on Test Set)

ii) Confusion Matrix of Keras-based Model (on Test Set)

**Supplementary Figure 1:** Feature Coverage and Class Coverage Percentage Plot

*A high-resolution version of the figure can be found as a supplement file “supp Fig-1.png.”*

**Supplementary Figure 2:** Distribution of Presence of Binary Features in Training, Validation and Testing Data

*A high-resolution version of the figure can be found as a supplement file “supp Fig-2.png.”*