Graph-based featurization methods for classifying small molecule compounds

Randy Posada, Mary Silva, Marisa Torres, Jonathan Allen, Jeff Drocco, Sarah Sandholtz, Adam Zemla, UCSF *SPOKE* investigative teams

University of California, Merced

Lawrence Livermore National Laboratory

Abstract

For over a decade, drug-induced liver injury (DILI) has posed significant drawbacks in the synthesis and development of drugs and remains a consequential concern. With finite success within the existing preclinical models, DILI is one of the main causes of drug withdrawal or termination from the market. Particularly, this withdrawal occurs during the late stages of drug development (Kullak-Ublick, 2017). Since DILI is difficult to diagnose and treat, it has become an obstacle in the drug production market that in turn affects clinicians, pharmaceutical companies, and consumers. We propose a method for learning features of DILI-positive drugs based on the graphical relationships and patterns they possess within a network of biological databases. We also train various statistical and machine learning models on these learned features in order to classify the drugs as DILI-positive or negative. Our methods include Random Forest, Neural networks, and logistic regression classification. We utilize labeled DILI-positive and DILI-negative datasets, which were developed by the FDA and the National center for toxicological research, as well as additional literature datasets (Thakkar, 2020) in order to validate our results and assess our featurization and model accuracy.

Keywords: liver toxicity, hepatoxic drug analysis, drug classification, FDA clinical trials, graph databases, data processing, graph embeddings, classification models, machine-learning featurization, model comparison.

1. Introduction

An important first step in identifying drug toxicity is to identify patterns in known liver toxic compounds. Current datasets are designed based only on FDA drug-labeling information, case registry-based approaches, or clinical evidence-based approaches (Thakkar, 2020). We hope to introduce additional datasets which allow us to capture patterns based on graphical relationships between compounds and known diseases, safety proteins, and genes. We begin by introducing the sources of the expanded datasets.

1.1 UCSF SPOKE

Provided by the University of California San Francisco, SPOKE (Scalable Precision Medicine Oriented Knowledge Engine) is a biological database of databases, offering researchers the ability to explore complex interconnected pathways. The graph-theoretic database harnesses the potential to enable discoveries and its diverse database continues to expand with the perpetual efforts to incorporate more information.

The SPOKE database is referred to as a heterogeneous network since it contains different nodes and edges that can each represent differing data and their corresponding connections. Through node-arc graphs, we can best observe the complex biological relationships that are present in diseases, treatments, and health. In this case, the “nodes” make up important human factors such as proteins, genes, anatomies, compounds, cell type, etc. Likewise, the “arc” or “edges” between the node pairs are representative of the known connections and relationships that exist between them. These node-arc relationships allow scientists to observe paths and follow a series of edges that may connect to nodes not previously known to have any relationship with one another.

Over two million compounds are mapped into SPOKE using “Smile Strings,” which are a way to describe a compound two-dimensionally. Smile strings are then matched using a Tanimoto similarity (or Jaccard) coefficient T, as a similarity measure to compare the chemical structure utilizing fingerprints. Further, the similarity, or proportion of the features shared among two compounds, is divided by their union to calculate the Tanimoto similarity. If the Tanimoto similarity result is greater than 0.85 then the two compounds or structures are considered similar. This cutoff is important because it helps us to create our eventual DILI/no-DILI labels to train the models.

1.2 Graphical Representation: Acetaminophen Example

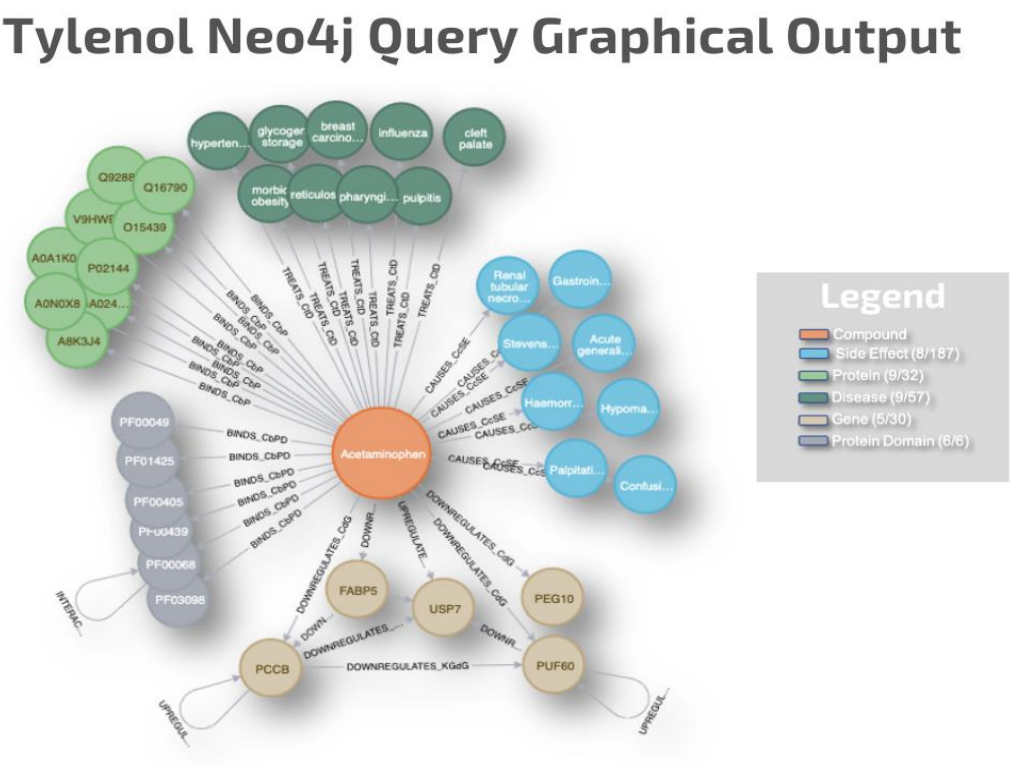
As our tool to visualize the SPOKE data, Neo4j stems far beyond the traditional table structure found in relational databases. It is a querying language similar to SQL, however, it contains flexible schemas for data storage and retrieval while representing the spatial relationships we see in Node-arc graphs. These observable node-edge relationships are of high importance when applying statistical analysis to the SPOKE data.

We can access the data stored in SPOKE through Neo4J queries, which produce a graphical result that allows us to observe a multitude of relationships. These queries can help us understand what is available within SPOKE and what edge connections are present.

The graphical output seen in figure 1 was obtained from a Neo4J query that was run for the compound “Acetaminophen,” ($C_8H_9NO_2$). Acetaminophen is one of Tylenol’s active ingredients. This compound is one of 2,093,045 compounds that comprise the SPOKE Database. Within the graphical representation, we can visualize the genes, proteins, side effects, and diseases, respectively as nodes, that are connected to, or have a relationship with our sampled compound.

FIGURE 1.

A Graphical Neo4J Output of For the Acetaminophen



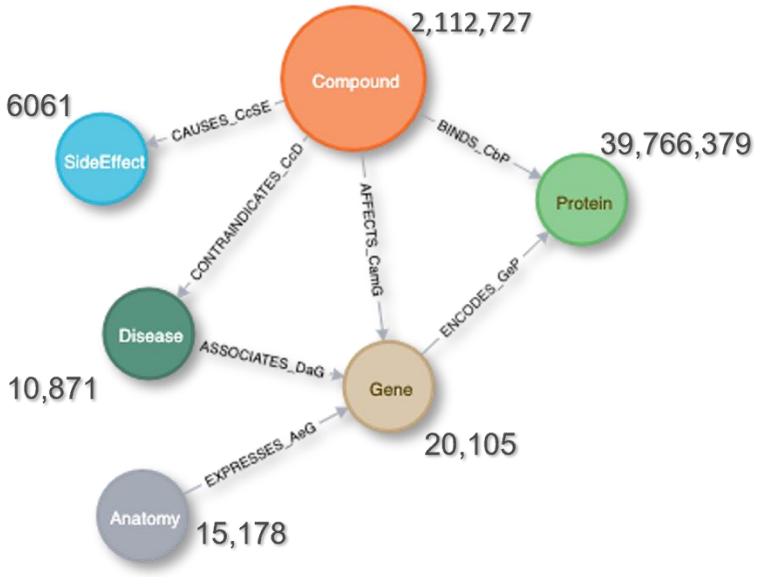
Note: The compound "Acetaminophen" is located at the center of the graph (Orange), and a small portion of the neighbor nodes are filtered out for visualization purposes and represents one observation of a compound subgraph.

The full graph contains more than twenty types of nodes, with over two million total observations. Each of these nodes has its properties, which range from a single numeric property to five or more properties of mixed types including descriptions, categories, sources, and more. Fifty-seven different edge types connect each of the node types, and over thirty million individual relationships will, at times, possess edge properties as well. We only consider a subset of nodes and edges in order to develop our baseline metrics.

In Figure 2, we present a graphical representation of a small portion of the SPOKE Database. Displayed next to each node, is their respective quantity count. We can observe the stature of this database by observing the compound quantity at 2,093,045 unique compounds. With this, these queries pose a computational rigor with increased run times due to the size of the data.

FIGURE 2.

A Visual Summary of Nodes and Edges Used for Modeling



Note: There are over two million compounds in the *SPOKE* database. This is a quantifying visualization of the node proportions within the continually updating dataset.

2. Methods

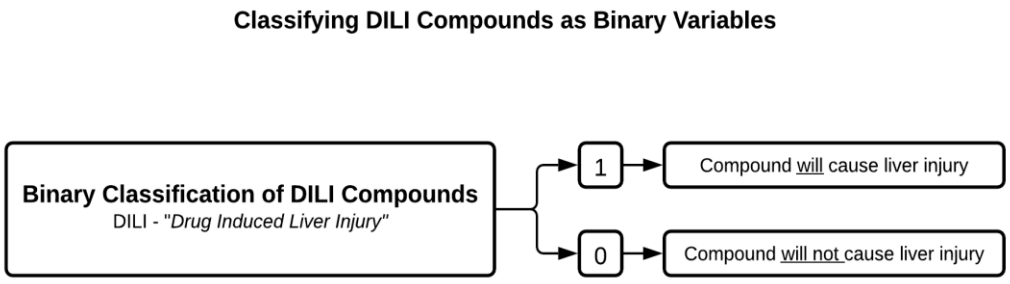
For an in-depth and inclusive analysis, we implement two approaches for model training, prediction, and inference. In the first approach, we create categorical features out of the node and edge relationships that are found within the graphs. We then perform statistical model fitting and prediction and compute the accuracy. We also explore a more recently developed graphical algorithm, the Node2Vec algorithm (Grover, 2022). In our efforts, we aim to measure the

computational rigor and model accuracies of each model. We compare a standard featurization and the Node2Vec algorithm to observe which approach offers the preferred computational efficacy and accuracy that is needed to screen compounds.

2.1 Baseline Featurization

Combined study data gave rise to the binary labels of either 0 or 1: the drug compounds found in the SPOKE database either do or do not injure the liver (Thakkar, 2020). This is characterized as 1 for 'will cause liver injury' or 0 for 'will not cause liver injury (Figure 3). This characterization is useful to identify if certain compounds within the featurized data have a likelihood of causing injury to the liver. Synthesizing new drugs can be facilitated by properly screening these compounds so that we know which compounds are labeled as "safe to use." This safety is determined through tests on either live animals or assays.

FIGURE 3. Using Binary Response Variables to Classify DILI Compounds for Featurization



Note: This visual provides a summary of the binary classification of the response, drug induced liver injury compounds, that were performed on our data from *SPOKE*.

We focused on only a small subset of data since the primary goal was to set up a baseline. We aim to build a model for general clustering or classification in subsequent studies. However,

for the purpose of this study, we use a small subset of labeled data from the total two million nodes and edges.

To better observe and compare the compounds from the SPOKE database, we formulate a tabularized feature matrix (Figure 4). Here we can observe the *Max Phase*, *Proportion liver anatomy*, *Proportion safety proteins*, alongside binary responses for *DILI*, *(Liver) Diseases*, and *(Liver) Side Effect*.

The compound data was incorporated into the SPOKE database from ChemBL, an open large-scale bioactivity database where data is regularly extracted from medicinal chemistry literature (Gaulton et al., 2017). The compounds possess unique ChemBL IDs, which allow researchers the ability to track and identify the bioactive molecules with drug-like properties and their chemical composition.

The *Max Phase* column, provided by ChemBL and the U.S. National Library of Medicine, indicates the maximum phase of clinical trials that a certain compound is on. A max phase of 0 indicates that the compound is currently in preclinical trials, while a max phase ranging from 1-3 indicates current clinical trials. The highest max phase a compound can achieve is a 4, which indicates that the compound has been approved as safe to treat diseases.

The proportion of anatomies that are connected to the compound through the node and edge relationships can be seen in the *Proportion liver anatomy* column. As seen on the fractional depiction in the first row, Acetaminophen possesses a total of ninety-two anatomy nodes on the subgraph, two of which are liver-related anatomies. Meanwhile, the *Proportion safety proteins* column indicates the proportion of compound experimental binding to proteins that may indicate potential drug safety issues or toxic effects.

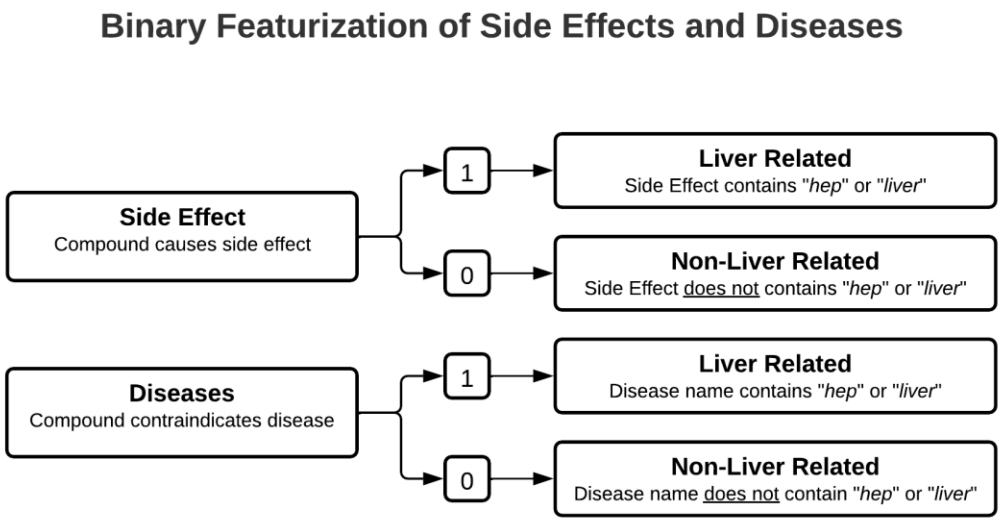
TABLE 1. ***Tabular representation of the feature matrix for the first six rows.***

Compound	Response	Max Phase	Proportion liver anatomy	Proportion safety proteins	(Liver) Diseases	(Liver) Side Effect
CHEMBL112 (Acetaminophen)	1	4	0.0217391304 (2/92)	0 (0/32)	1	1
DB00322	1	4	0.02173913	0	1	0
DB00768	1	4	0.021978022	0.1	0	0
DB01136	1	4	0.02173913	0.0390625	0	1
CHEMBL1406871	0	0	0.025316456	0	0	0
CHEMBL3800326	0	0	0.022727273	0.033333333	0	0

Note: A random sample of six rows from our feature matrix are tabularized above to provide a visual the variables used after pre-processing. The first compound is Acetaminophen (CHEMBL112), as shown in Figure 1.

We create binary vectors from our categorical variables by applying a binary featurization for the two features: “Side effect” and “Diseases.” The feature "Side Effect" is one-hot encoded as 1 for liver-related side effects, and 0 for non-liver-related side effects. The feature "Diseases" is one-hot encoded as 1 for liver-related disease association and 0 for non-liver-related disease association (Figure 5). With the features defined, we may apply any existing machine learning or statistical models, which we describe later in this section as part of the evaluation.

FIGURE 4. ***Visualizing the conducted binary featurization of spoke data*.**



Note: Similar to figure 3, we visually depict the conversion of SPOKE data to a categorical (binary) form that allows the model to interpret both the Side Effects and Diseases.

2.2 Node2Vec Featurization

We also use a graphical algorithm to create a feature representation with an embedding algorithm known as Node2Vec. Node2Vec is a node algorithmic framework for learning continuous feature representations in latent space. This algorithm computes a continuous vector representation of a node based on random walks in the graph. The multiple samples, or 'walks,' are taken beginning with the *Compound* node and traversing, or 'stepping,' through the *SideEffect*, *Disease*, and *Gene* node types in our graph. The sampling is limited to these node types due to the memory size limitations, but future subgraph filtering could solve this. These 'random walk' samples are then treated as 'words in a document' in the sense that traditional Word2Vec embedding algorithms can be applied to the contents of each document (Mikolov, 2022). In this

analogy, the ‘words’ are the contents of the nodes, and the overall corpus is the combined collection of samples. The idea is that similar compounds have similar ‘documents’, and the embedding hopefully captures these similarities. Other document modeling procedures can also be considered in the future.

The end result is an n-dimensional, numeric representation of each compound. These n-dimensional vectors then become the feature inputs for existing machine learning or statistical models. In our example, we use a 20-dimensional feature representation that scales to the amount of ‘walks’ performed in the random walk. This parameter could also be optimized using various statistical procedures, but for the purposes of this study, we are only establishing a baseline.

2.3 Evaluation

In order to evaluate the performance of our algorithms, we must utilize clearly labeled and well-studied liver toxicity data. To begin, we randomly sampled compounds and their subgraphs from SPOKE and ensured that the DILI and No-DILI response is balanced. As our train/test split, we used eighty percent of the data to train the models and the other twenty percent to test them.

To obtain a well-performing and accurate model we use multiple simple classification models to establish a baseline for featurization, training, and testing. We create a logistic regression model, a neural network/multi-layer perceptron, and a random forest classifier.

As a classification algorithm, we apply logistic regression to model the probability that a certain class or event exists. In our case, we are interested in learning the probability that a drug induces liver injury or the probability that it does not induce liver injury. We can use this model to predict a binary outcome, such as zeros or ones, given a set of independent variables. This model can predict the probability of occurrence of an event by fitting data to a logit function.

For comparison, we also use a neural network classification algorithm commonly used in machine learning, the multi-layer perceptron (MLPClassifier). For our study, our neural networks consist of one input layer, a single hidden layer, and binary outputs.

The third model we implement is a random forest classifier that takes the average of many classification decision trees from random sampling for a refined model. For our study, 1000 trees were obtained for sampling.

3. Results

We compare both the one-hot-encoding baseline featurization method and Node2Vec featurization methods in terms of computational efficiency and prediction accuracy.

3.1 Computational Efficiency

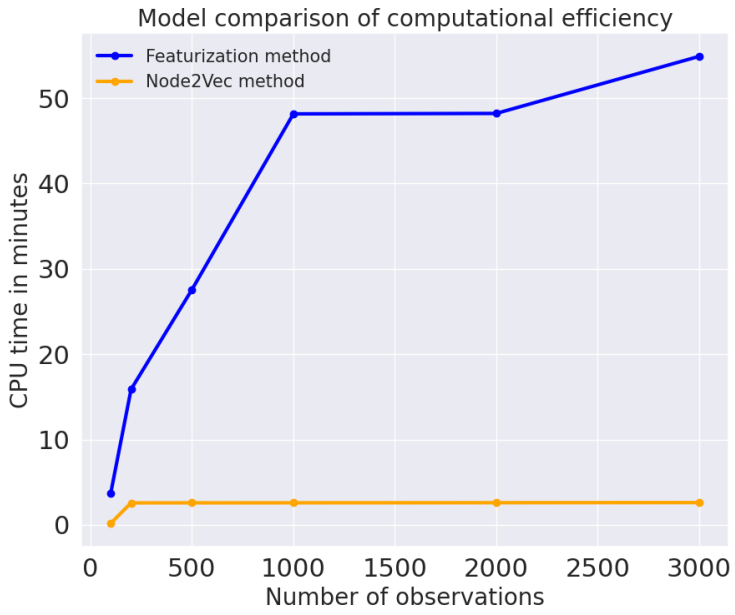
For the one-hot-encoding featurization process, sampling the subgraph and encoding when a relationship does or does not exist is the most exhaustive step in the featurization. Depending on how many individual compounds we have, and how many outgoing relationships they share, then checking for a keyword or character match, the computation time increases exponentially with each observation.

On the other hand, for our Node2Vec algorithm, since we have the database stored in memory and only care about the sampling of each node, the only value that is important in this process is the node's unique identification number. Once the node IDs are sampled, we can then map those IDs back later once the embedding is obtained. As a result, once embeddings are created for all the nodes in the graph, the remaining computation time for any random sample is limited to model fitting and predicting. Additionally, the complexity of Node2Vec does increase depending on how many steps and how many walks from each compound we decide to use. The

more random walk samples we have, the longer the embedding process, but the fitting and prediction process is less affected by compound sample size.

The featurization method increases exponentially as more observations are added. Meanwhile, the second and more efficient method remains at a constant performance under ten minutes (Figure 6). By comparison, it would take the featurization method more than fifty minutes of CPU time to compute while the Node2Vec method would only utilize three minutes, thus proving a greater computational time efficiency per observation count.

FIGURE 5. ***Comparing computational efficiency by observation compute time***



Note: The model comparison denotes the rigorous amount of computational time that the featurization takes in comparison to the much more efficient Node2Vec method.

3.2 Prediction Accuracy

We tabularize the prediction accuracies for our train test split amongst two different sample sizes. For visualization purposes, we display a smaller sample to compare to the larger sample of 1000 compounds. For most models, we can observe a notable difference as more samples are

incorporated into the model, the smaller the prediction accuracy. Our greatest observed accuracy was for the random forest classifier using the Node2Vec featurization which holds an 80% accuracy. The lowest accuracy resulted from the neural network model using the baseline featurization (Table 1).

TABLE 2. ***Prediction accuracies for the data at various sample sizes.***

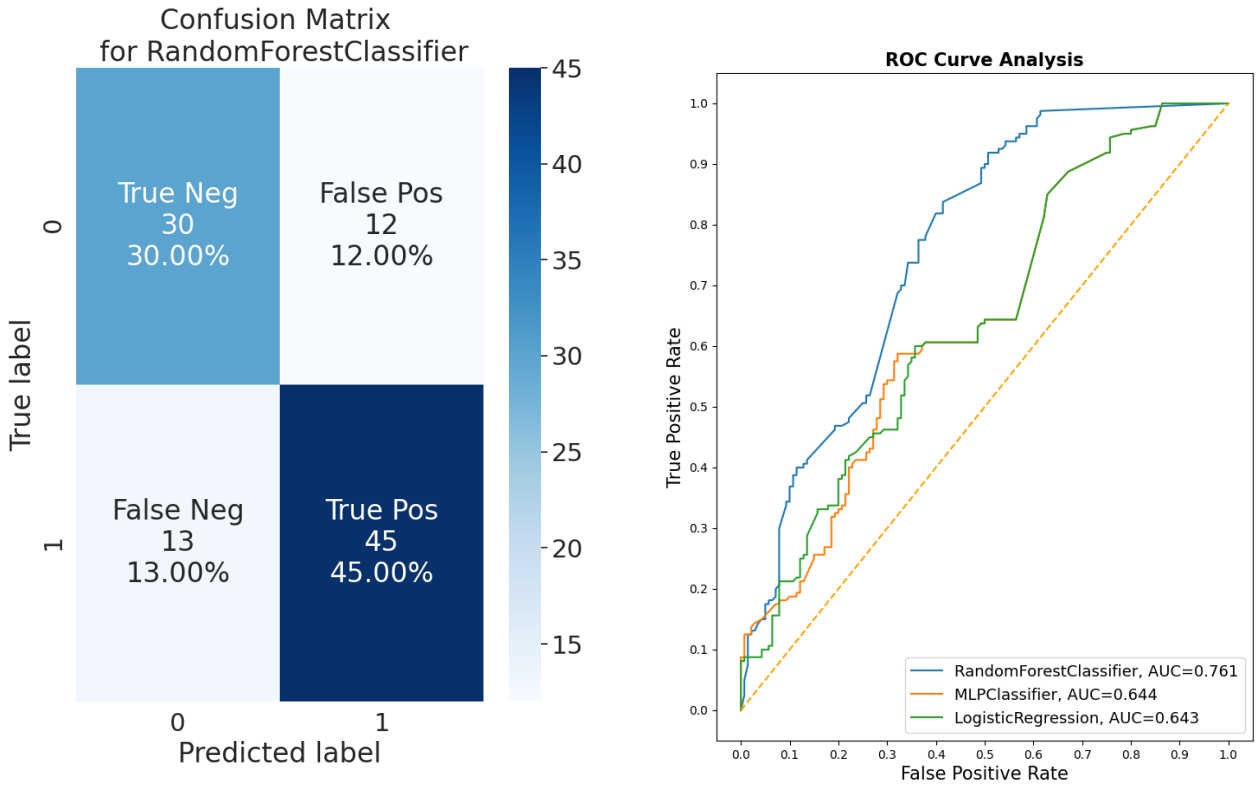
		Accuracy	
Sample Size	**Model**	**Approach 1 (Baseline Featurization)**	**Approach 2 (Node2Vec)**
500	Random Forest	67.3%	80%
	Neural Network	54.3%	77%
	Logistic Regression	55.6%	76%
1000	Random Forest	66.8%	75%
	Neural Network	61.3%	76.5%
	Logistic Regression	56.5%	70.5%

Note: The table depicts the recorded prediction accuracy for a 80%-20% train/test split. The percentage represents the percent correctly predicted using the corresponding model.

The confusion matrix in Figures 7 and 8, shows the false positive vs true positive for the best performing classifier models using our baseline featurization approach and Node2Vec embeddings, respectively. True positive and true negative is relatively high, but not perfect. The results of the models are analyzed within a Confusion matrix, which portrays a tabular representation of actual versus predicted values. This helps us to find the accuracy of the model and avoid overfitting by analyzing the percentages associated with its quadrants.

In the same figures, we also conducted ROC Curve Analysis for the baseline featurization and Node2Vec embeddings. For this metric, we look for the curve with the largest AUC (Area Under Curve). An ideal curve would showcase the true positive rate being high and the false positive rate being low, leading to a curve that closely hugs the upper left corner. Therefore, we want to choose a model whose curve is closest to the parameters of a perfect classifier.

FIGURE 6. ***Visual results for the model fitted to the baseline featurization***



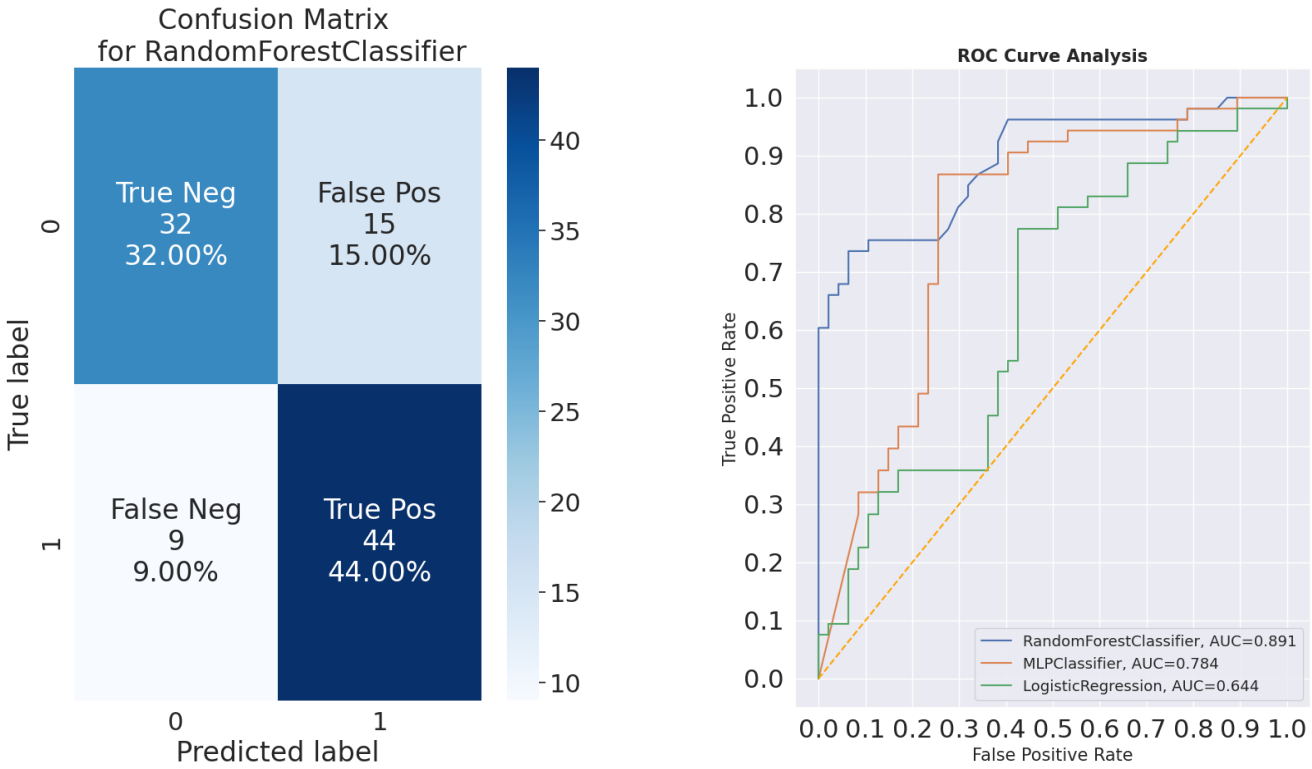
Note: The confusion matrix (left) and ROC curve analysis (Right) for the test set after fitting the random forest classifier on the baseline featurization method.

3.3 Summary of Featurization results

After tabularizing our featurization results (Table1), we observe that within the baseline featurization approach (column: Approach 1), for both sample sizes of compounds, the Random

Forest method yields better results. We determine the random forest classifier has the highest accuracy in the baseline featurization method. The ROC curve analysis indicates that the random forest classifier model using the baseline featurization outperforms the other models used at an AUC of 76.1% (Figure 7).

FIGURE 7. ***Visual results for the model fitted to the Node2Vec embedding***



Note: The confusion matrix (left) and ROC curve analysis (Right) for the test set after fitting the random forest classifier on the Node2vec featurized data.

3.4 Summary of Node2Vec results

Within the same table, using a sample size of only 500 compounds, the Node2Vec column (Approach 2) identifies the random forest as the best model at an 80% prediction accuracy (Table 1). For a sample size of 1000, the prediction accuracy of the Node2vec embedding decreases to

76% for the neural network model. The ROC curve for the random forest model using the Node2Vec embedding featurization is calculated to have an AUC of 89% (Figure 8).

4. Conclusion

From our obtained results we state that the approach using the Node2Vec algorithm is both computationally faster and more accurate when using a subset of labeled liver toxicity data. It is important to note that both featurization methods decrease in prediction accuracy as the sample sizes, or the number of compounds included in our model, increases.

As we increase the number of compounds, we also increase their interconnected neighbor nodes. These neighbors might contain information that does not necessarily help our model and ends up hindering our computational efficiency. For example, if we included a compound that is unrelated to liver toxicity, it may have node neighbors that do indicate safety proteins or liver and hepatic side effects. This ultimately confuses the model or forces it to be categorized as DILI or no-DILI when it is potentially neutral. In this instance, a multi-class model would be required but is beyond the scope of this study.

In subsequent studies, we aim to explore how we can filter, or assign a preference to certain nodes during the Node2Vec random walk-in order to capture better embeddings. For instance, we want the random walk to traverse neighbor nodes and are ‘hep’ or ‘liver’ related with higher priority. We’d also like to better optimize the embedding parameters, such as how many random walks per compound, and how many steps for each walk. The number of embedding dimensions should also be considered. Ultimately, intending to establish a baseline, our approaches performed well. Any additional feature representations can be compared to this baseline to determine if the prediction accuracy and efficiency improve.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract Contract DE-AC52-07NA27344 (LLNL-JRNL-832728).

References

Chen, M., Vijay, V., Shi, Q., Liu, Z., Fang, H., Tong, W. "FDA-Approved Drug Labeling for the Study of Drug-Induced Liver Injury," Drug Discovery Today, 16(15-16):697-703, 2011

David, S., & Hamilton, J. P. (2010). Drug-induced Liver Injury. US gastroenterology & hepatology review, 6, 73–80.

Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, Davies M, Dedman N, Karlsson A, Magariños MP,

Overington JP, Papadatos G, Smit I, Leach AR. (2017) 'The ChEMBL database in 2017.' Nucleic Acids Res., 45(D1)

Grover, A., Leskovec, J., (2022) Node2vec: scalable feature learning for networks. node2vec. (n.d.). Retrieved February 17.

Kullak-Ublick, G. A., Andrade, R. J., Merz, M., End, P., Benesic, A., Gerbes, A. L., & Aithal, G. P. (2017). Drug-induced liver injury: recent advances in diagnosis and risk assessment. Gut, 66(6), 1154–1164.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September 7). Efficient estimation of word representations in vector space. arXiv.org. Retrieved February 17, 2022

Thakkar, S., Li, T., Liu, Z., Wu, L., Roberts, R., & Tong, W. (2020). Drug-induced liver injury severity and toxicity (DILIst): binary classification of 1279 drugs by human hepatotoxicity. Drug Discovery Today, 25(1), 201–208.