Multiclass disease predictions based on integrated clinical and genomics datasets

Multiclass Disease Predictions Based on Integrated Clinical and Genomics Datasets

Moeez M. Subhani
College of Engineering and Technology
University of Derby
Derby, England
Email: m.subhani@derby.ac.uk

Ashiq Anjum
College of Engineering and Technology
University of Derby
Derby, England
Email: a.anjum@derby.ac.uk

***Abstract*—Clinical predictions using clinical data by computational methods are common in bioinformatics. However, clinical predictions using information from genomics datasets as well is not a frequently observed phenomenon in research. Precision medicine research requires information from all available datasets to provide intelligent clinical solutions. In this paper, we have attempted to create a prediction model which uses information from both clinical and genomics datasets. We have demonstrated multiclass disease predictions based on combined clinical and genomics datasets using machine learning methods. We have created an integrated dataset, using a clinical (ClinVar) and a genomics (gene expression) dataset, and trained it using instance-based learner to predict clinical diseases. We have used an innovative but simple way for multiclass classification, where the number of output classes is as high as 75. We have used Principal Component Analysis for feature selection. The classifier predicted diseases with 73% accuracy on the integrated dataset. The results were consistent and competent when compared with other classification models. The results show that genomics information can be reliably included in datasets for clinical predictions and it can prove to be valuable in clinical diagnostics and precision medicine.**

Keywords–Clinical; Genomics; Data Integration; Machine Learning; Disease Prediction; Classification; Bioinformatics.

I. INTRODUCTION

The medical science is rich with various types of datasets ranging from clinical to genomics datasets. The clinical datasets are diverse in terms of their nature, format and the information they contain. On the other hand, genomics datasets are intrinsically enormous in size and dimensions, and so is the information contained in them [1]. The genomic information can be considered as the backbone of clinical information since the genomic structure derives the physical characteristics of any organism. If the two pieces of information are connected, it may help to improve the overall medical research by finding more accurate and advanced clinical diagnostic solutions. The connection essentially means to integrate clinical and genomics datasets. This is also a way forward in precision medicine studies, where medical practitioners want to make clinical decisions based on both clinical and genomics parameters and not just one of them [2][3].

However, the research to establish or explore this connection is not very commonly sought in the state-of-the-art [1][4]–[6]. The datasets from clinical and genomics sources are mainly used independently in their respective research domains. From literature review, it has been observed that most clinical prediction studies have been limited to either clinical datasets [7]–[13] or genomics datasets [14]–[20]. One common factor among these studies is that almost all of them are prediction studies, which establishes the fact that the trend for clinical predictions has long prevailed in research.

Although there are some studies which have attempted towards the inter-domain research, the trend does not seem to be very progressive. For example, [21] used decision trees to predict breast cancer outcomes. Similarly, [22] employed multiple regression and statistical methods to infer associations, and [3] used a graph-based approach to predict cancer clinical outcomes from multi-omics data. All these studies used integrated datasets for prediction or association studies using various approaches. However, most of these approaches are now outdated due to limitations in terms of their performance or accuracy [21][22]. The approach in [3] (combination of regression, Bayesian networks, and evolutionary neural networks) is more advanced and promising but this study is limited to binary classifications and multi-omics data only [23][24].

The research work mentioned above show that prediction based studies are common in the literature. The most popular or commonly sought predicting factors are survival rate and disease recurrence rate. However, we could not find any disease prediction model in the literature based on combined clinical and genomics data information. A typical disease prediction model, as we define, takes information from both clinical and genomics datasets and predicts disease(s) in a patient. This can be achieved when we have both clinical and genomics datasets available for a variety of diseases. Hence, we are attempting to design a disease prediction model which aims to predict possible medical condition(s) in a patient using information from both clinical and genomics datasets.

From ClinVar and Expression Atlas databases, we have been able to construct such dataset which contains both clinical parameters as well as gene expression values in a single dataset for several patients. Since the data retrieved from these databases is in eXtensible Markup Language (XML) format, we can create a very flexible schema for this dataset. Using this dataset, we can train a model to learn the diseases in various subjects. As an initial attempt to prove the concept, we have used the k-Nearest Neighbours (kNN) algorithm for the learning model, which is an instance based learner [25]. Considering the size and complexity of the dataset, kNN appears to be a reasonable choice of learning method since it learns the classification function only locally.

Genomics based clinical diagnosis does not exist in clinical environments. Traditionally, disease predictions are made using regular clinical practices only. Our disease prediction model can provide a genomic signature to verify the disease existence or possible occurrence. Hence, this model not only will help

 ISBN: 978-1-61208-717-7

TABLE I. CLINVAR DATASET.

GENE	CONDITION	CLINICAL SIGNIFICANCE	CHROMOSOME NO.	LOCATION	VARIATION ID	ALLELE ID
AKAP	LONG QT SYNDROME	BENIGN/LIKELY BENIGN	7	92001306	136347	140050
AKT2	COLORECTAL NEOPLASMS	LIKELY PATHOGENIC	19	40236313	376039	362918
APC	HEREDITARY CANCER-PREDISPOSING SYNDROME	PATHOGENIC	5	N/A	181836	181126
...	...	...	...	...	...	...

the medical practitioners to gain another step of confidence in terms of clinical diagnosis, but also help advance the precision medicine research.

The rest of the paper is arranged as follows. Section II discusses the challenges for data integration. Section III explains the data integration model. Section IV gives details of the prediction model and the algorithm along with the implementation details. Section V presents the results, followed by discussion in Section VI and conclusion in Section VII.

II. CLINICAL AND GENOMICS DATA INTEGRATION CHALLENGES

The integration of clinical and genomics datasets is crucial to move towards precision medicine. The medical conditions of each person are transcribed from the underlying genomics structure. Hence, it is critical to bring forward the genomic information to play part in the clinical diagnostics [2][3]. The main challenge is to find a way to integrate datasets which are completely different from each other in terms of their nature, size, and properties.

Most biological databases have standardised the data storage in XML formats. European Molecular Biology Laboratory (EMBL) took an initiative in 2000 to provide access of all the flat files data in XML format [26]. XML provides more flexibility in terms of storage, transport and integration of complex biological datasets [27]. The format also provides the advantage that the schema of datasets is extensible and multiple datasets can be mapped together. Our datasets from both sources, ClinVar and Expression Atlas, are accessed in XML formats.

The scope of data integration models is vast, as mentioned in the literature review in the previous section. Various data integration models have been discussed by various authors including [1], [4] and [6]. For our study, we have adopted a meta-dimensional approach model, which refers to using multiple datasets simultaneously in the analysis [6]. This involves building a model on top of multiple datasets, which are combined or integrated either before or after building the data model. The approach facilitates the advantage of fetching information from multiple datasets and including it in the analysis model. However, the integration may also yield complex datasets resulting in less robust models.

There are multiple methods within the meta-dimensional approach as mentioned by [1] and [6]. We have adopted a concatenation-based integration method, where different matrices are combined into a large single matrix before building a model. One advantage of this method is that once it is determined how to concatenate the variables from different datasets into a single matrix, it is relatively easier to build any statistical analysis model on it. For example, on a combination of genomics datasets, [8] used a Bayesian model to predict phenotypes, and [28] used Cox Lasso model to predict time to recurrence.

It may be important to mention here that the integration attempt in this paper is only at the data level. Since the data being retrieved from public repositories is in XML format, we do not need to pre-build a structure to store data, and we are not dealing with databases either. Therefore, this method provides the advantage to avoid the data structure and storage issues. Hence, the data integration here must not be confused with the traditional database level data integration.

TABLE II. GENE EXPRESSION DATASET.

GENE	GSM452573	GSM452571	GSM452642	...
AKAP9	3.563587736	3.45243272	3.535150355	...
AKT1	10.8863402	10.34918494	9.129441853	...
AKT2	5.005896122	4.463927997	4.993673626	...

III. DATA INTEGRATION MODEL

We have used completely anonymised clinical and genomics datasets obtained from public sources. The clinical dataset (ds1) has been obtained from ClinVar [29], which is an open source database that contains information about the genomic variation and links it with phenotype information. For each gene, it provides the diseases it causes and their clinical significance. In addition, it also includes the whereabouts of the gene, such as, chromosome number, location, variation ID etc. A snapshot of the data is illustrated in Table I. The database was searched for 'colorectal cancer', and all the search results were downloaded and saved as XML files.

The genomics dataset (ds2) is a Gene Expression dataset of primary colorectal tumours (E-GEOD-18105), obtained from the Expression Atlas of European Bioinformatics Institute (EBI), which is a public resource for gene expression datasets [30]. Gene expression data, as the name indicates, contains information for the expression of gene(s) in a particular biological sample(s). The expression data is obtained via microarray technology, which provides parallel processing and monitoring

 ISBN: 978-1-61208-717-7

TABLE III. INTEGRATED DATASET.

DISEASE	CLINICAL SIGNIFICANCE	CHROMOSOME NO.	LOCATION	VARIATION ID	ALLELE ID	GENE	GSM452573	GSM452571
LONG QT SYNDROME	BENIGN/ LIKELY BENIGN	7	92001306	136347	140050	AKAP	3.563587736	3.45243272
COLORECTAL NEOPLASMS	LIKELY PATHOGENIC	19	40236313	376039	362918	AKT2	5.005896122	4.463927997
...	...	...	...	...	...	...	...	...

of tens and thousands of genes, producing tons of valuable data [31]. A typical gene expression dataset contains a matrix with genes in rows and samples in columns. The number in each cell of the matrix characterises the expression level of a specific gene in the given sample [32]. Table II shows an example of how gene expression data looks like. After the first column, which is gene name, the rest of the columns represents samples, and the values represent the expression levels.

The primary reason for selecting these two datasets was that they fulfill the information requirement for this study. The ClinVar data provides the information about clinical condition against each gene present in the dataset. It also provides the clinical significance of these conditions [33]. The gene expression data brings the information about the activity of those genes in different samples. Hence, the two datasets provide the required information to create an integrated dataset for this model.

TABLE IV. STATISTICS OF INTEGRATED DATASET.

Output Classes	I	II
Unique Classes	80	76
Feature set	117	117
Training Examples	258	281

As mentioned previously, we are using a meta-dimensional approach based integration, and specifically the concatenation method. The datasets were concatenated via gene names. It has to be noted that there were multiple examples for each gene in both datasets. The examples in the ds1 with no feature sets available in ds2 were removed. On the contrary, the examples in the ds2 for which there were no feature sets in ds1, the data was extrapolated in ds1 so that the examples for that gene can be increased. Since each parameter in the feature set is independent, therefore, extrapolating some points does not affect the accuracy.

Table III shows an example of the integrated dataset, where the clinical and genomics parameters are concatenated via gene names. The statistics of the dataset is shown in Table IV. The data was trained with two different output classes: genes (class-I) and diseases (class-II). There are 80 unique genes, and 76 unique diseases in the dataset after removing the outliers.

It can be argued that predicting genes as output class does not provide much meaning. Predicting disease has a more clinical value since this information is not available in the gene expression data. The reason behind this selection is only to provide an example that the classifier can be used to predict any feature from an integrated dataset without any restriction.

The resulting schema includes clinical and genomics parameters in columns, while each row represents a gene. Hence, each row tells the possible medical condition for a gene if it is active in a sample. This schema is completely flexible and scalable. It can be expanded by adding data from different sources, as long as the new data can be mapped to existing schema. More data brings more information that will only help to improve the performance of the classifier by increasing the feature set and the training examples.

IV. PREDICTION MODEL

In this section, we will talk about the multiclass classification challenges, followed by the details of our prediction model, comprising the algorithm and the experimental environment.

A. Multiclass Classification

When we talk about disease classification, we are talking about a complicated multiclass classification problem. From classification perspective, it is relatively easier to classify binary problems or even few classes, but with increasing number of classes, the complexity of the dataset gets very high [34]. The data under consideration in this study contains more than 75 different classes. When the number of output classes is that high, the variance in the data is very high as well. In such a case, it is best to have as much data as possible so that every class has a sufficient representation in training data. This is a minor limitation in our study because of the limited number of examples available from public datasets.

There is no single classification method that can be suggested to be best suited for multiclass classification [34]. Any algorithm can perform better than the rest based on the characteristics and properties of the data. In this study we have used the k-Nearest Neighbours (kNN) algorithm. The reason for selecting the kNN instead of Support Vector Machines (SVM), which is a more popular classification algorithm, is our large number of output classes and the random distribution of data (Figure 1). Unlike SVM, which uses kernels for optimization, kNN determines the label for a given data point based on nearest data points on the distance metric. Since kNN is a non-parametric algorithm, it does not assume any explicit functions for the input data (such as Gaussian) [25]. This works well in our case when the data has no particular distribution and is widespread (Figure 1). Hence, we can avoid the algorithmic complexity by using an algorithm which uses

 ISBN: 978-1-61208-717-7

local optimization only. Also, kNN performs well on small to medium sized datasets [25].

B. Classification Algorithm

The kNN is a non-parametric supervised learning algorithm [25]. For a given dataset X, with labels Y, the algorithm calculates the distances between a new data point z and all data points in X to create a distance matrix. Euclidean distance is the most common method for calculating this distance. Euclidean distance between point x_i and y_i can be calculated by:

$$D(x, y) = \sum_{i=1}^{k} (x_i - y_i)^2 \quad (1)$$

Let $R = (X_i, Y_i)$, where $i = 1, 2...N$, be the training set, where X_i is the $p * q$ feature vector, and Y_i is the q-dimensional vector which represents m output class labels, as we are considering multiclass classification problem. We presume that the training data has random numeric variables with unknown distribution.

From the training set R, the kNN algorithm narrows down to a local sub-region $r(x)$ of the input space, which is centered on an estimation point x. This predicting sub-region $r(x)$ contains the training points (x') nearest to x, which can be expressed as:

$$r(x) = \{x' \mid D(x, x') \leq d(k)\} \quad (2)$$

where, $D(x, x')$ is the distance metric between x' and x, and $d(k)$ is the k^{th} order statistic. $k[y]$ denotes the k samples in the sub-region $r(x)$, which are labelled y. The kNN algorithm estimates the posterior probability $p(y \mid x)$ of the estimation point x:

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} \cong \frac{k[y]}{k} \quad (3)$$

Generally, when the kNN is used for binary classification, the label assignment is relatively easier since the algorithm has to select between two classes only, such as :

$$g(x) = \{\, 1\ , k[y = 1] \geq k[y = -1] - 1, k[y = 1] \leq k[y = -1] \quad (4)$$

We have improvised this functionality for our study, where the output class is non-binary. In this case, for any estimation point x, the decision $g(x)$ for a given label y is estimated by:

$$g_k(x) = y_k \mid minD_k \quad (5)$$

where, D_k is represented by 1 Hence, the decision that will maximise the posterior probability will be assigned for the output label. For a multiclass classification problem, where $y \in \{1 \ldots k\}$, the kNN algorithm uses the following decision rule:

$$F(x) = argmax[g_k(x)] \quad (6)$$

Thus, for the selected nearest k neighbours, the algorithm calculates the posterior probability for each class, and the class with highest probability is assigned to x. Euclidean distance is the most common method, but there are other distance calculation methods as well, such as seuclidean, mahalanobis, spearman, etc [25].

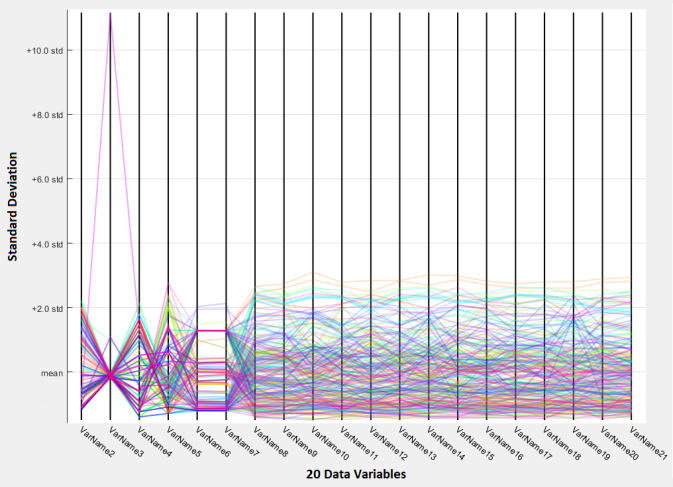

Figure 1. Distribution of variance in the integrated dataset.

C. Performance Measurement

Generally, the performance of machine learning classifiers is measured using various parameters, such as, accuracy, sensitivity, specificity, and Receiver Operator Curve (ROC). These parameters are calculated based on the true positives, true negatives, false positives and false negatives of classifier. For binary classes, these parameters are easier to calculate because there is only one positive and one negative class. However, for multicalss classification, the problem is more complicated and it is not easy to calculate each parameter for each class. Especially ROC, which is a standard measure to represent performance of a classifier, is very complicated to calculate for a very large multiclass problem. This problem has been discussed in further detail by Fawcett in [35].

Therefore, calculating each parameter for every class will not only be laborious, but will also produce loads of results that will be difficult to ensemble and explain. To simplify that, we have only used confusion matrices to represent the performance of the classifier and used the accuracy for each classifier to compare the results for the two classes.

D. Experimental Environment

We have used Matlab (R2018a) for all the experiments, which provides built-in libraries for machine learning classifiers. We used the machine learning toolbox to train the classification model using kNN. The toolbox takes the data as input and process the classification itself using the built-in library functions and selected features. The classification toolbox uses the Euclidean distance by default to compute the

 ISBN: 978-1-61208-717-7

distance metrics. The tool box can be used to reproduce the results.

At first, we perform the Principal Component Analysis (PCA) for dimensionality reduction. Since, our data is multivariate, ranging from gene expression data to phenotypic data, the data points are widespread in the data space. Figure 1 shows the standard deviation distribution of the first 20 data variables from the integrated dataset. It can be seen that the data distribution is very random and does not follow any standard distribution function. Therefore, it is important to reduce the dimension of the integrated dataset. We performed PCA to explain 95% variance in the data.

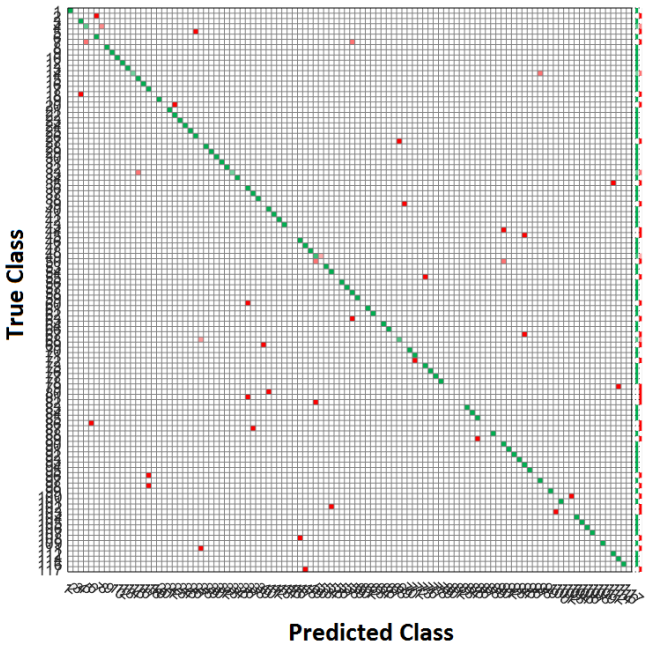

Figure 2. Confusion matrix for class-I.

The results of classification depend highly on the dimensions of the dataset. The correlation between the number of examples and feature sets is very critical in this case to avoid over-fitting [36][37]. The clinical dataset has only 5 features, which is not a large enough set to be used stand-alone for prediction model. With a feature set of 5, the prediction is neither reliable nor comparable with other datasets. The genomics dataset is large enough in this respect, but it does not contain the class-II so we cannot predict diseases. Therefore, we have only used the integrated dataset to train with the prediction model explained in the previous section (IV), and then compared it with other classifiers.

The results are validated using 10-fold cross-validation. This means, the dataset is divided into 10 parts; one part is held out as a test data and the rest of the 9 parts are used as training data. This step is repeated 10 times using a different part every time to holdout as a test data. This way every example from data is used both as training and test data. The resulting accuracy is an average of the 10-fold process.

V. RESULTS

The performance of a classification model is analysed using a confusion matrix. Figure 2 shows a confusion matrix for class-I prediction. The rows in a confusion matrix represent the true output class, and the columns represent the predicted class. The diagonal cells indicate the true positives (green) and the false negatives; and the off-diagonal cells indicate the false positives and the true negatives (red). The bottom right cell shows the overall accuracy and the loss of the classifier.

A. Classification with our Classifier

The number of neighbours (NN) is a variable in the algorithm, which can be tuned to change the performance of the algorithm. We tested the performance of the algorithm over 10 different neighbours, from 1 to 10.

As mentioned previously, we trained the integrated dataset for two different classes: genes (class-I) and diseases (class-II). The results are shown in Figure 3. At NN=1, the trained model predicts class-I with 86% accuracy, and class-II with 73% accuracy.

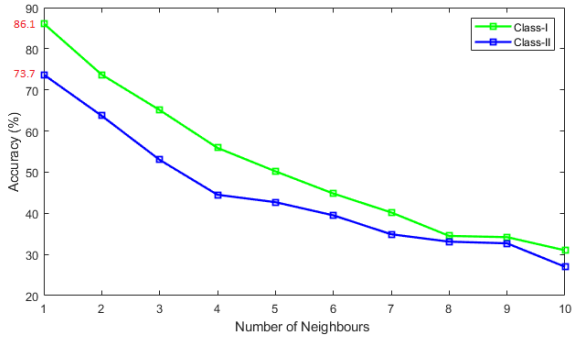

Figure 3. Accuracy of classification model for both classes.

Initially, the accuracy drops almost linearly with the increasing number of neighbours. The drop in accuracy can be attributed to the variation in data. As the algorithm considers more number of neighbours, each neighbour brings more variation that affects the prediction accuracy. However, as it can be seen in Figure 3, accuracy remains above 50% for up to 3 neighbours for both classes which can be regarded as a good accuracy considering a multivariate training data. Following NN=4, the accuracy drops almost exponentially.

This variation over neighbours may be avoided by introducing a weighted parameter in the algorithm. This parameter weighs the contribution of each neighbour under consideration based on its distance. The nearest neighbours gets higher weights than the distant ones. Matlab's classification tool uses the squared inverse method to calculate the weights, which can be expressed as:

$$w_n = \frac{1}{d(x_n - x_i)^2} \tag{7}$$

where, x_n is the neighbour to point x_i. To accommodate this weight parameter, the eq1 is adjusted as follows:

 ISBN: 978-1-61208-717-7

$$D(x,y) = \sum_{i=1}^{k} w_i (x_i - y_i)^2 \quad (8)$$

We tested this updated version by training the integrated dataset, and we observed that the accuracy was raised to the maximum (86.1% for class-I and 73.7% for class-II) for all NN's. The results are shown in Figure 4. This is perhaps because the weighted version predicts based on the neighbour with the highest weight. Since the nearest neighbour is most likely to have highest weight out of all neighbours, the classification result is the same every time. This result seems to be not very helpful for our dataset.

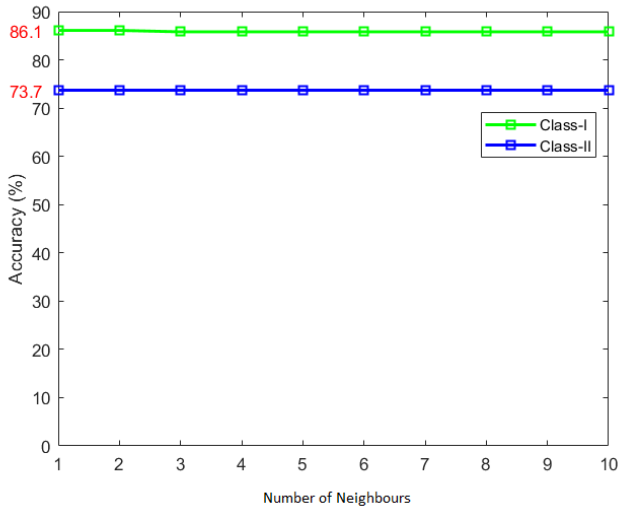

Figure 4. Accuracy for both classes with weighted kNN.

However, our model has predicted the diseases with up to 73% accuracy. The accuracy is not as good as for the class-I (86%). There can be multiple reasons behind this. The representation of each class label in the data varies, which affects the prediction accuracy. Some classes have sufficient examples in the data, while others have only few examples. The higher the representation of a class label in the training data, the better is the prediction accuracy for that class. The distribution of class-I labels in dataset is comparatively more uniform than class-II; hence, higher accuracy. Still, achieving 73% accuracy for class-II is a very good result considering the size, shape, and multivariate nature of the dataset.

B. Comparing with other Classifiers

We trained the same integrated dataset with other classifiers in order to compare the performance. Using PCA of 95%, we trained all the classifiers available on Matlab's classification toolbox, and then selected bluethe top 10 models (out of 22) to compare the classification accuracy for both classes. NN=1 for all the models in the classification toolbox. 10-fold cross-validation was used to avoid over-fitting. The results are shown in Figure 5.

For class-I, the kNN models provided the highest accuracy of 96.9%. kNN was followed by the Tree and SVM models. As

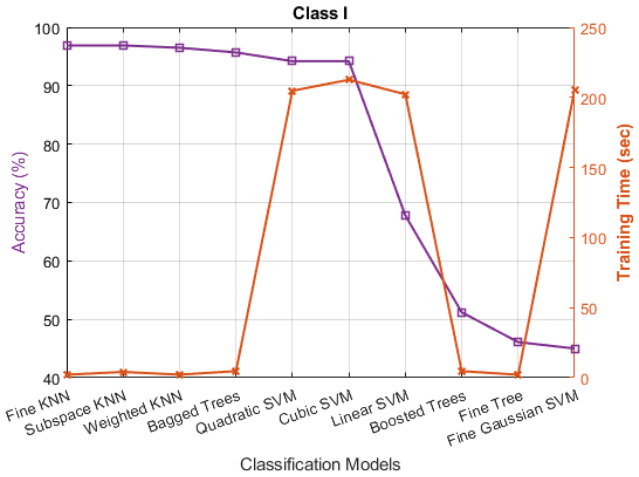

Figure 5. Performance of other classification models for class-I predictions.

we can see, the top three models are all kNN models providing accuracy of above 95%. The accuracy of the SVM models (Quadratic and Cubic) is almost in the same range (95-96%), however, the training time for the SVM models is 200 times higher than the kNN models. This is because the SVM uses the cost minimization functions, such as gradient descent or kernel functions, which take much longer to converge. Since kNN does not use any of those functions, it is more robust and provides with the same, rather better accuracy. To summarise, although both kNN and SVM models have predicted about the same accuracy, the kNN models are much more robust than the SVM models in terms of performance.

The tree models, except for bagged trees, performed poorly providing accuracy of about 50% or under. The training time of the tree models is as good as that of kNN models (few seconds), but the accuracy is poor. Bagged trees, which is a bootstrapping method, performed quite well. On the other hand, boosted trees provides an accuracy of just about 51%. Although both of them are ensemble methods, which means they provide an average of multiple models trained on a subset of data, bagged trees provided much better result.

The accuracy of these kNN models (Fine kNN, Subspace kNN, and Weighted kNN) is slightly higher than our prediction model (Figure 3). The reason for this is that the models in the toolbox are set on different defaults and use different functions than the ones we used. The classification function that we used is primarily for multiclass classification problems. On the other hand, the function used by the toolbox models are mainly designed for binary problems, hence, the difference of accuracy.

Similar results are seen for class-II. The results are shown in Figure 6. The top 10 models selected here are slightly different than those for class-I, but majority are the same. The highest accuracy achieved for class-II is 73.3%, which is just about the same as achieved by our model (Figure 3). The top 3 models are all kNN models, with bagged trees standing at 4th position with 73% accuracy. All SVM models provide accuracy of less than 50% with training times as high as over 200 times of the kNN models. The same is the case for tree

 ISBN: 978-1-61208-717-7

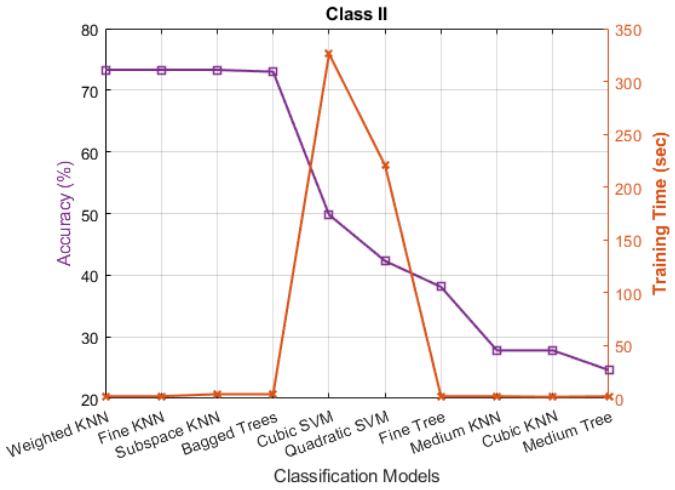

Figure 6. Performance of other classification models for class-II predictions.

models except for bagged trees; same result as for class-I. A plausible explanation for good performance of bagged trees could be that they perform better on high dimensional data.

VI. Discussion

We have demonstrated a novel way for multiclass classification based on integrated clinical and genomics datasets. We have used concatenation-based data integration model for this purpose, which has been discussed by various researchers before ([1][6]), but not implemented in the area of health care. Hence, this is the first time that we have attempted to use this meta-dimensional approach to integrate datasets.

In the past, people have used various other methods for data integration such as tree-based models [21], statistical models [22], and graph based models [3][18][20]. All these models require considerable amount of effort and time to build the data models first, before creating the data analysis model, such as building the binary trees, or creating graphs models from datasets. Our method does not involve any of those complex models; it only requires concatenation of all the datasets into a single matrix. Once concatenated, the model transfers the dataset directly to the analysis model and starts training the learning algorithm. Hence, it is way more efficient in terms of time and computational costs as compared to other methods.

In terms of analysis, from our knowledge, none of the previous models have been used for multiclass disease classification problems in health care. They have only been demonstrated for binary classifications; and, therefore, their results cannot be compared with our model, which is a multiclass classification model.

In terms of data models, it will be very difficult to perform multiclass classification based on the previously mentioned models because they will require to build a separate data model (trees of graphs) for each output class before the analysis model. Having multiple output classes, the analysis models will get extremely complicated with several input data models. With our proposed model, as there is only single concatenated dataset, the multiclass classification is less complicated and manageable because the dataset has only one data model with a single schema.

Since, we could not compare our results with any other previous results from other researchers, we have demonstrated comparison with other classification models. The results shown in Figures 5 and 6 demonstrate that the kNN models can outperform the rest of the classification models in terms of prediction accuracy and performance.

Our proposed approach provides a very flexible and scalable model, along the lines of our previous work as reported in [38]–[41], which can be scaled to adjust any new dataset and accommodate any analysis model. As long as there is a relational dataset, it can be concatenated to the existing dataset within the same data model and schema. Any analysis model or algorithm, including prediction, classification, regression models, can be built on top of the dataset. This flexibility enables this approach to be adapted for any research purpose in any domain.

VII. Conclusion and Future Directions

The way forward in precision medicine is to use all available data from clinical and genomics domains in order to provide the best clinical solutions. The datasets need to be intelligently integrated for this purpose. In this paper, we have performed clinical predictions based on clinical and genomics information. We have attempted to integrate a clinical (ClinVar) and a genomic (gene expression) dataset, and performed classification for disease predictions. We have designed a multiclass classification model that predicts diseases from integrated datasets. The model, which is validated by 10-fold cross-validation, has predicted diseases with up to 73% accuracy. We also predicted genes as an extra variable, from the same dataset, and achieved up to 86% accuracy. We have compared the results with other classification models and demonstrated that our model outperforms the rest. We can conclude that constructing the learning classifiers on top of large-scale inter-domain integrated datasets can provide very good clinical predictions. This can prove to be very beneficial and a stepping-stone towards the precision medicine.

This research study shows that diseases can be predicted with good accuracy from a patient's dataset if it has both clinical and genomics parameters present. The accuracy will further improve if we train the model with a much larger size of training data. The reliability and confidence in results will increase by incorporating more clinical and genomics information. We have demonstrated with a gene prediction example, that, when the dataset is more uniformly distributed among different classes, the prediction accuracy goes high even on a multiclass classification task.

This study has great potential to expand including achieving analysis provenance [13]. The more information a dataset will contain, higher the accuracy can be achieved. The dataset can be expanded to include more multivariate clinical and genomics datasets, such as clinical trials and multi-omics datasets, respectively. Including clinical information from clinical trials or laboratory tests will have a significant impact in the clinical prediction studies.

 ISBN: 978-1-61208-717-7

ACKNOWLEDGMENT

The authors would like to thank and acknowledge the help and support received from Usman Yaseen, Sanna Aizad, Bilal Arshad and Craig Bower in the successful completion of this study.

REFERENCES

[1] M. M. Subhani, A. Anjum, A. Koop, and N. Antonopoulos, "Clinical and genomics data integration using meta-dimensional approach," in 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC). IEEE, 2016, pp. 416–421.

[2] R. Higdon et al., "The promise of multi-omics and clinical data integration to identify and target personalized healthcare approaches in autism spectrum disorders," Omics: a journal of integrative biology, vol. 19, no. 4, 2015, pp. 197–208.

[3] D. Kim et al., "Knowledge boosting: a graph-based integration approach with multi-omics data and genomic knowledge for cancer clinical outcome prediction," Journal of the American Medical Informatics Association, vol. 22, no. 1, 2014, pp. 109–120.

[4] J. S. Hamid, P. Hu, N. M. Roslin, V. Ling, C. M. Greenwood, and J. Beyene, "Data integration in genetics and genomics: methods and challenges," Human genomics and proteomics: HGP, vol. 2009, 2009.

[5] B. Louie, P. Mork, F. Martin-Sanchez, A. Halevy, and P. Tarczy-Hornoch, "Data integration and genomic medicine," Journal of biomedical informatics, vol. 40, no. 1, 2007, pp. 5–16.

[6] M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, and D. Kim, "Methods of integrating data to uncover genotype–phenotype interactions," Nature Reviews Genetics, vol. 16, no. 2, 2015, p. 85.

[7] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: a comparison of three data mining methods," Artificial intelligence in medicine, vol. 34, no. 2, 2005, pp. 113–127.

[8] B. L. Fridley, S. Lund, G. D. Jenkins, and L. Wang, "A bayesian integrative genomic model for pathway analysis of complex traits," Genetic epidemiology, vol. 36, no. 4, 2012, pp. 352–359.

[9] D. Kim, R. Li, S. M. Dudek, and M. D. Ritchie, "Athena: Identifying interactions between different levels of genomic data associated with cancer clinical outcomes using grammatical evolution neural network," BioData mining, vol. 6, no. 1, 2013, p. 23.

[10] T. A. Lasko, J. C. Denny, and M. A. Levy, "Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," PloS one, vol. 8, no. 6, 2013, p. e66341.

[11] R. Palaniappan, K. Sundaraj, and S. Sundaraj, "A comparative study of the svm and k-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals," BMC bioinformatics, vol. 15, no. 1, 2014, p. 223.

[12] J. Raikwal and K. Saxena, "Performance evaluation of svm and k-nearest neighbor algorithm over medical data set," International Journal of Computer Applications, vol. 50, no. 14, 2012.

[13] R. McClatchey et al., "Providing traceability for neuroimaging analyses," International journal of medical informatics, vol. 82, no. 9, 2013, pp. 882–894.

[14] U. D. Akavia et al., "An integrated approach to uncover drivers of cancer," Cell, vol. 143, no. 6, 2010, pp. 1005–1017.

[15] M. P. Brown et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," Proceedings of the National Academy of Sciences, vol. 97, no. 1, 2000, pp. 262–267.

[16] D. Kim, H. Shin, Y. S. Song, and J. H. Kim, "Synergistic effect of different levels of genomic data for cancer clinical outcome prediction," Journal of biomedical informatics, vol. 45, no. 6, 2012, pp. 1191–1198.

[17] L. J. Lu, Y. Xia, A. Paccanaro, H. Yu, and M. Gerstein, "Assessing the limits of genomic data integration for predicting protein networks," Genome research, vol. 15, no. 7, 2005, pp. 945–953.

[18] E. E. Schadt et al., "An integrative genomics approach to infer causal associations between gene expression and disease," Nature genetics, vol. 37, no. 7, 2005, p. 710.

[19] J. Zhu et al., "Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks," Nature genetics, vol. 40, no. 7, 2008, p. 854.

[20] X. Yang et al., "A network based method for analysis of lncrna-disease associations and prediction of lncrnas implicated in diseases," PloS one, vol. 9, no. 1, 2014, p. e87797.

[21] J. R. Nevins, E. S. Huang, H. Dressman, J. Pittman, A. T. Huang, and M. West, "Towards integrated clinico-genomic models for personalized medicine: combining gene expression signatures and clinical factors in breast cancer outcomes prediction," Human molecular genetics, vol. 12, no. suppl_2, 2003, pp. R153–R157.

[22] E. Lee, S. Cho, K. Kim, and T. Park, "An integrated approach to infer causal associations among gene expression, genotype variation, and disease," Genomics, vol. 94, no. 4, 2009, pp. 269–277.

[23] D. Kim, R. Li, S. M. Dudek, and M. D. Ritchie, "Predicting censored survival data based on the interactions between meta-dimensional omics data in breast cancer," Journal of biomedical informatics, vol. 56, 2015, pp. 220–228.

[24] S. S. Verma, D. Kim, M. D. Ritchie, A. Lucas, R. Li, and S. M. Dudek, "Using knowledge-driven genomic interactions for multi-omics data analysis: metadimensional models for predicting clinical outcomes in ovarian carcinoma," Journal of the American Medical Informatics Association, vol. 24, no. 3, 2016, pp. 577–587.

[25] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification and scene analysis. Wiley New York, 1973, vol. 3.

[26] A. Robinson, J.-J. Riethoven, and L. Wang, "XEMBL: distributing EMBL data in XML format," Bioinformatics, vol. 18, no. 8, 2002, pp. 1147–1148.

[27] Z. Lacroix, "Biological data integration: wrapping data and tools," IEEE Transactions on Information Technology in Biomedicine, vol. 6, no. 2, 2002, pp. 123 – 128.

[28] P. K. Mankoo, R. Shen, N. Schultz, D. A. Levine, and C. Sander, "Time to recurrence and survival in serous ovarian tumors predicted from integrated genomic profiles," PLOS ONE, vol. 6, no. 11, 2011, pp. 1–12.

[29] "NCBI ClinVar Database," URL: https://www.ncbi.nlm.nih.gov/clinvar [accessed: January 2018].

[30] "Gene Expression Atlas," URL: https://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-18105/?query=E-GEOD-18105_A-AFFY-44 [accessed: January 2018].

[31] N. Raghavachari, "Microarray technology: basic methodology and application in clinical research for biomarker discovery in vascular diseases," in Lipoproteins and Cardiovascular Disease. Springer, 2013, pp. 47–84.

[32] A. Brazma and J. Vilo, "Gene expression data analysis," FEBS Letters, vol. 480, no. 1, 2000, pp. 17–24.

[33] M. J. Landrum et al., "Clinvar: public archive of relationships among sequence variation and human phenotype," Nucleic acids research, vol. 42, no. D1, 2013, pp. D980–D985.

[34] C. Zhang, M. Ogihara, and T. Li, "A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression," Bioinformatics, vol. 20, no. 15, 2004, pp. 2429–2437.

[35] T. Fawcett, "An introduction to roc analysis," Pattern recognition letters, vol. 27, no. 8, 2006, pp. 861–874.

[36] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty, "Optimal number of features as a function of sample size for various classification rules," Bioinformatics, vol. 21, no. 8, 2004, pp. 1509–1515.

[37] A. K. Jain and W. G. Waller, "On the optimal number of features in the classification of multivariate gaussian data," Pattern recognition, vol. 10, no. 5-6, 1978, pp. 365–374.

[38] A. Anjum, R. McClatchey, A. Ali, and I. Willers, "Bulk scheduling with the diana scheduler," IEEE Transactions on Nuclear Science, vol. 53, no. 6, 2006, pp. 3818–3829.

[39] F. van Lingen et al., "The clarens web service framework for distributed scientific analysis in grid projects," in 2005 International Conference on Parallel Processing Workshops (ICPPW'05). IEEE, 2005, pp. 45–52.

[40] F. Van Lingen et al., "Grid enabled analysis: architecture, prototype and status," 2005.

[41] S. L. Kiani, A. Anjum, M. Knappmeyer, N. Bessis, and N. Antonopoulos, "Federated broker system for pervasive context provisioning," Journal of Systems and Software, vol. 86, no. 4, 2013, pp. 1107–1123.

 ISBN: 978-1-61208-717-7