Empowering Multi-class Medical Data Classification by Group-of-Single-Class-Predictors and Transfer Optimization: Cases of Structured Dataset by Machine Learning and Radiological Images by Deep learning

Tengyue Li, Simon Fong, Sabah Mohammed, Jinan Fiaidhi, Steven Guan and Victor Chang*

***Abstract*—In the medical domain, data are often collected over time, evolving from simple to refined categories. The data and the underlying structures of the medical data as to how they have grown to today's complexity can be decomposed into crude forms when data collection starts. For instance, the cancer dataset is labeled either benign or malignant at its simplest or perhaps the earliest form. As medical knowledge advances and/or more data become available, the dataset progresses from binary class to multi-class, having more labels of sub-categories of the disease added. In machine learning, inducing a multi-class model requires more computational power. Model optimization is enforced over the multi-class models for the highest possible accuracy, which of course, is necessary for life-and-death decision making. This model optimization task consumes an extremely long model training time. In this paper, a novel strategy called Group-of-Single-Class prediction (GOSC) coupled with majority voting and model transfer is proposed for achieving maximum accuracy by using only a fraction of the model training time. The main advantage is the ability to achieve an optimized multi-class classification model that has the highest possible accuracy near to the absolute maximum, while the training time could be saved by up to 70%. Experiments on machine learning over liver dataset classification and deep learning over COVID19 lung CT images were tested. Preliminary results suggest the feasibility of this new approach.**

***Index Terms*—Machine learning, Deep learning, Multi-class classification, Parameter optimization, Classification model training, Medical dataset, Radiological images recognition, algorithm.**

I. Introduction

Medical data are usually collected over time, and the data schema might have evolved from data that are composed of simple to refined categories. For example, when medical records about cancer disease were initially collected, they may only be labeled either benign or malignant in the simplest form. As medical knowledge about the disease advances by medical discovery or better electronic patient record technology becomes available, the features in the dataset expand from embracing binary class to multi-class. More labels of sub-categories of the disease are added accordingly. However, the same complex data are made up of several classes of data, putting together as a multi-class dataset. This multi-class dataset is possible to be decomposed back to several subsets, each of which only contains certain binary classes of data.

What factors should be considered when determining whether to use multiple binary classifiers or one multiclass classifier for such data in medical informatics? Perhaps creating a complex multiclass classifier is not the best option. Alternatively, if we want multiple binary classifiers to work together, a general strategy is similar to the One-vs-All set. In One-vs-All, you essentially have an expert binary classifier that is really good at recognizing the pattern from everyone else, and the implementation strategy is usually cascading. For example, we could have a quatro-class classification model with classes Normal, Class A, Class B, and Class C. Figure 1 shows an illustration of a One-vs-all example:

This work was supported in part by 2018 Guangzhou Science and Technology Innovation and Development of Special Funds, via Grant no. EF003/FST-FSJ/2019/GSTIC, and code no. 201907010001, and also VC Research (VCR 0000149).

Tengyue Li is with the Data Analytics and Collaborative Computing Laboratory, University of Macau, Macau SAR (yb97475@um.edu.mo).

Simon Fong is with the Department of Computer and Information Science, University of Macau, Macau SAR (e-mail: ccfong@umac.mo).

Sabah Mohammed is with the Department of Computer Science, Lakehead University, Thunder Bay, Canada (e-mail: mohammed@lakeheadu.ca).

Jinan Fiaidhi is with the Department of Biotechnology, Lakehead University, Thunder Bay, Canada (e-mail: jfiaidhi@lakeheadu.ca@lakeheadu.ca).

Steven Guang is with Xl'an Jiaotoing-Liverpool University, Suzhou, China (e-mail: Steven.Guan@xjtlu.edu.cn)

Victor Chang* is with Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK. (e-mail: v.chang1@aston.ac.uk and ic.victor.chang@gmail.com) and corresponding author

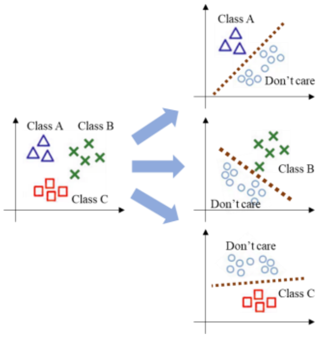

Fig. 1. Multiple one-versus-all binary-class classifiers

One approach is to build a model that does multi-category of disease classification. In a very simple example, there are four outputs from a disease classifier:: [None, Class A disease, Class B disease, Class C disease]. There are two ways to approach this: One option uses a multi-class classifier.

Multi-class classifier: [None, Class A disease, Class B disease, Class C disease]

Another option is to use multiple binary classifiers such as follow:

Single-classifier A: [None, Class A disease]
Single-classifier B: [None, Class B disease]
Single-classifier C: [None, Class C disease]

Distinguishing which is a better option and in which condition the option works better is not easy. Multi-class classifiers have the following advantages and disadvantages. The advantages are: Easy to use out of the box since there is only one model to deal with, and is convenient when you have many classes in the dataset. The disadvantages are usually slower than binary classifiers during training; they could really take a while to converge for high-dimensional problems. Some popular multi-class classifiers are Tree-based algorithms and artificial neural network type of algorithms.

1.1 Motivation

One-vs-All classifiers' advantages and disadvantages are as follow. The advantages are simplicity and fast convergence are usually resulted from binary classifiers. It is good and perhaps transparent pertaining to interpretable or explainable AI that is good to have a handful of individual classifiers that offer the probability of how the prediction of classes comes about. However, the disadvantages may be cumbersome to deal with when you have too many classes unless a systemic model is available. Training individual classifiers over subsets of data may lead to class imbalance-related problems that lead to bias, e.g. if you have a large number of samples of none and few samples of a particular disease type, or vice-versa. Some popular methods are most ensemble methods, support vector machines and pruning-enabled tree algorithms that trim off tree branches biased towards a majority class.

In One-vs-All, you essentially have an expert binary classifier that is good at recognizing one pattern from all the others, and the implementation strategy is typically cascaded.

Although the one-vs-all classification concept has been around for some years, it did not gain the popularity as deserved, probably due to some cons. The limitations that we observe from one-vs-all and the corresponding solutions we propose are as follows.

1.2 Contribution

A novel strategy called Group-of-Single-Class prediction (GOSC) coupled with majority voting and model transfer is proposed for achieving optimally maximum accuracy at only a fraction of the required long training time. The main advantage is the ability to achieve an optimized multi-class classification model that has the highest possible accuracy near to the absolute maximum, while the training time could be minimized.

Since many individuals and independent binary-class classifiers are hard to handle, a solution is to have an ensemble-like methodology to harness a collection of expert binary-class classifiers. This is the principle of Group-of-Single-Class prediction (GOSC). Each expert binary-class classifier delivers a single-class SC prediction, with probability scores and resulting rules explaining the outcomes. The final prediction is inferred by majority voting, which logically evaluates and selects the most probable result from the most reliable model. A reliable model is deemed one with high composite performance over several essential indicators such as accuracy, kappa, false-alarm rate, balanced precision and recall, etc.

Furthermore, after optimization, each SC classifier (SCC) shares and copies its best model configuration within the GOSC framework. The best model configuration of a SCC would be transferred to the construction of a multi-class classifier. This is similar to transfer learning in deep learning terminology. It helps spare the time-consuming model optimization for the multi-class classifier. As a result, the users can opt to use either a near-optimal multi-class classifier or majority voting of a group of SCC's. This novel methodology is suitable for medical informatics based on the assumptions that medical records are built over time, from binary-class to increasingly complex multi-class add-ons.

The remainder of this paper is organized as follows. The related work in Section 2 reviews similar one-vs-all classification examples that have been applied prior and followed by our proposed methodology, namely GOSC in Section 3. The experiment is conducted and described in Section 4. The discussion of the experiment results is presented in Section 5. Section 6 concludes the paper.

II. Related Work

Related works to the aspects of the growth of complexity in medical data, hence the motivations of this research, and some background of transformation from multi-class to binary class classifiers are reviewed in this section.

As the electronic Health Record technology matures, the complexity of the data grows in several directions. Developments of new clinical cases enable more data features to be added in describing the data. The increasing ease and advances of data collection and big data archiving techniques give rise to data volume. Cloud computing and online platforms enable PACS to fuse multiple data sources more easily than ever. For example, [1] reported multiple sources of data could be successfully aggregated onto a patient-centric health data-sharing platform.

Given the abundance of health data and disease-related data that grow in increasing complexity, scientists are eager to get on mining them for insights and discovery and building predictive models over the data for classification. A lot of research efforts were focused on single disease analysis. A great deal of machine learning was applied to analyze the data about a single disease. Uddin et al. [2] compared the performance of single disease prediction using a range of supervised machine learning algorithms. Ding et al. [3] investigated machine learning algorithms for predicting individual diseases. Nguyen et al. [4] proposed a special machine learning network model for predicting whether breast cancer will relapse or not, focusing on only a single disease and binary outcome.

Moreover, research progresses towards finding the relations and risks of how certain diseases might have occurred together in the same data record. Multi-label classification gained popularity, expanding the single horizon of a particular to considerations of multiple diseases or symptoms that simultaneously show up, e.g., a heart attack is associated with blood pressure, hypertension, obesity, healthy diet and diabetes, etc. Another complexity is a multi-class model that is built on data that encompasses multiple classes. The prediction target is not merely binary but spans several possible classes. This usually gives a more precise classification assessing which exactly is the characteristic or severity level of disease. For example, in cancer staging, there are four major levels, ranging from stage 0 that is a healthy body, to stage 1, where the cancer cells are small and confined in a small area, to stages 2, 3 and 4, which eventually spreads to other parts of the patient's body. Bayati et al. [5] invented an inexpensive method for multiple disease prediction.

Multi-class Classification (MCC) classifies the testing data into more than two classes, e.g., class A, class B and class C. Each data is labeled to one and only one class by MCC. Data can be classified into one of the classes A, B, or C, but at the same time cannot be both.

In the medical domain, numerous researchers report on predicting or analyzing the likelihood of having a single disease at a time. For diabetes analysis, Neuvirth et al. [9], Shivakumar et al. [8] and Yeh et al. [7] built models that classify a disease by the presence or absence of the disease. Likewise, for predicting the presence of cerebrovascular disease, the same type of models was constructed [7, 10]. Typically, binary classification takes care of the predictions of single diseases. Nevertheless, several related diseases may simultaneously occur where binary classification is insufficient to handle multiple classes effectively. Runzhi et al. [6] attempted to use an ensemble multi-label classification model for predicting the risks of multi-diseases from physical examination records.

In millennia, Allwein et al. [11] were pioneers in unifying from several simple classifiers into a multi-class classifier capable of handling multiple binary problems. The experimental results prove that their method offers a feasible alternative to the commonly used multi-class models, giving rise to the popular adoptions of support vector machines and AdaBoost. Dong et al. [2] extended the idea to a tree structure of nested hierarchical that replaces a multi-class model of multiple classes by individual binary-class models. The method generated random partitions of ensembles of sampling trees, and it is proven to be an effective approach in lieu of a multi-class model. The researchers also managed to fix the unbalanced binary class problems over the fact that the partitioned data may contain too much class-irrelevant data and too few class-specific data. Galar et al. [3] has extensively tested such concept of simplifying a multi-class to a number of binary-class models, calling them one-versus-one and one-versus-all, etc.

Galar and his team tested a number of popular machine learning algorithms from the literature, such as decision tree, SVM, IBL and rule-based methods. The results show the binarization approach, which decomposes a multi-class model to multiple binary-class models, has certain benefits. The results are verified by statistical significance analysis. It was found that the robust techniques include J48 (decision tree), JRip (rule-based method) and SVM have significant advantages when the multi-class problem is turned into binary-class models. But instance-based learning technique like kNN has little difference.

Fürnkranz [14] compared the one-versus-one decomposition techniques with respect to the suitability of decision tables and decision trees. The comparison is against popular ensemble methods such as bagging and boosting and bagging. The results indicate that an appropriate method for combining the outputs is needed to achieve performance improvement using confidence estimates.

Based on the relevant literature, it is confirmed that the prior works have shown breaking down a complex multi-class classifier into the binary class classifier. The advantage is observed from converting a single multi-class model into a group of binary-class models. However, the extent of advantages varies from algorithm to algorithm. We are inspired to assure the performance of grouping up binary-class models and using them as if they are one multi-class classification model. Another way is to balance the imbalanced class data after decomposition, just as the same problem was fixed in [2]. It is known that choosing the right parameters is crucial to which model performance is sensitive [15][16]. In light of ensuring a good level of accuracy for a classification model or algorithm, one aspect is to get the model parameters optimized, which are able to maximize the model performance. Hence, this becomes a motivation in this study to embrace parameter optimization as a part of an investigation of model binarization.

III. Proposed Methodology

Extending from Galar et al. [3], a methodology based on a Group-of-Single-Class prediction (GOSC) plus certain modifications are proposed. A traditional multi-class prediction methodology, in Figure 2a works by supervised learning a model from a training dataset that consists of multiple classes. Like a standard supervised learning process, the dataset of multiple classes is loaded into an induction process where training and validation occur, learning a model over the data. Once training is done, the model becomes mature, this multi-class classifier (MCC) is ready for making predictions by loading in some unseen testing data.

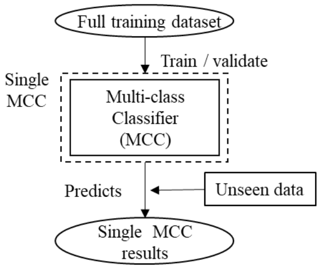

Fig. 2a. A traditional multi-class prediction methodology

By the design of the GOSC methodology, the model induction and testing process is expanded to three layers, as shown in Figure 2b. Firstly the full training dataset is split by the classes, partitioning it into multiple training datasets. Each has a group of data associated with a particular class label. The individual subset of data of a single class that has binary labels (existence of a particular disease versus non-existence of that disease) is used to train a single binary-class classifier. The number of single binary-class classifiers is equal to the number of classes. Prior to these binary-class classifier trainings, the data subset that is often imbalanced will be subject to rebalancing, using resample and/or SMOTE [17].

Each of the individual binary-class classifiers is subject to cross-validation parameter optimization called *CVParameterSelection* [18]. It performs parameter selection by cross-validation to find a set of parameters that give the best performance. It is known that optimal parameters yield optimal model configuration, therefore optimal performance for a classifier [19][20][21]. However, the searching process is tedious and time-consuming, especially if it were to be done by trial-and-error manually. *CVParameterSelection* selection function automates the search by testing out users‘ specified ranges of parameters that contribute to the base classifier's model setup. After the classifier model is optimized and the optimal parameters are found, the new model is ready for subsequent predictions. The level of prediction accuracy is supposed to improve as well [22].

With a group of binary-class classifiers optimized and ready to predict, a majority voting mechanism [23] is applied at the prediction phase. The full framework of the Group-of-Single-Class prediction methodology is shown in Figure 2b.

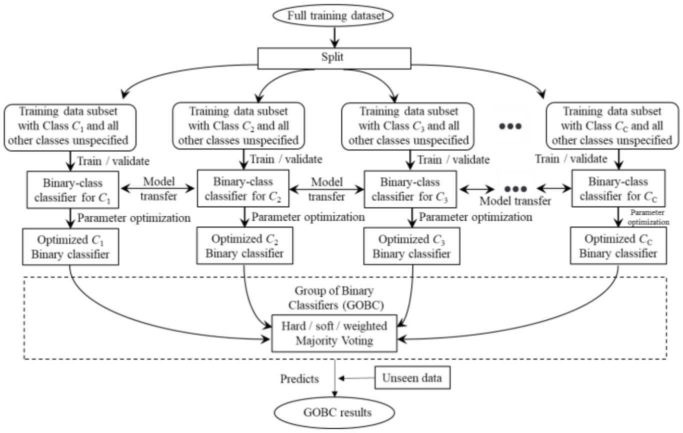
Full training dataset
Split
Training data subset with Class C_1 and all other classes unspecified
Training data subset with Class C_2 and all other classes unspecified
Training data subset with Class C_3 and all other classes unspecified
●●●
Training data subset with Class C_C and all other classes unspecified
Train / validate
Train / validate
Train / validate
Train / validate
Binary-class classifier for C_1
Model transfer
Binary-class classifier for C_2
Model transfer
Binary-class classifier for C_3
●●●
Model transfer
Binary-class classifier for C_C
Parameter optimization
Parameter optimization
Parameter optimization
Parameter optimization
Optimized C_1 Binary classifier
Optimized C_2 Binary classifier
Optimized C_3 Binary classifier
Optimized C_C Binary classifier
Group of Binary Classifiers (GOBC)
Hard / soft / weighted Majority Voting
Predicts
Unseen data
GOBC results

Fig. 2b. The framework of Group-of-Single-Class prediction methodology

Majority voting is a typical ensemble machine learning that uses a group of binary-class classifiers instead of an MCC [24]. For predicting a continuous future variable, such as forecasting or regression, the output from a voting ensemble will be the mean of the predicted results from the group of prediction models [25].

For classifying samples into discrete labels, as in our case of medical disease classification, a hard majority voting (MV) method is used [26]. The hard MV mechanism collects the outputs of all the binary-class classifiers, each classifier votes for a particular class. Hard MV collects the votes and selects a winning class with the most votes as the final prediction outcome. When two or more votes are in deuce, the tie is broken by judging from the classifiers' accuracy performance. The one that has the highest level of accuracy has the winning vote. The predicted probabilities for classes are summed up.in the soft MV method [26]. Then it predicts the winning class to which the sample should be classified by the class with the largest sum probability. Users could optionally choose between hard or soft MV methods to handle the group voting and make collective predictions from the individual binary-class classifiers.

One innovation in our methodology is the model transfer technique. The model transfer is referred to the concept of copying the optimal parameters from the best performing binary-class classifiers to the multi-class classifier prior to its training [27]. The motivation is to speed up the whole training process for MCC, which includes parameter optimization, model construction and n-fold cross-validation. The parameter optimization times are known to be extremely long for the multi-class model. In contrast, the parameter optimization times for binary-class classifier is much shorter. The imbalanced data due to splitting the original training sets into subsets of binary-class data need to be rebalanced. It is an important criterion to produce a good quality well-trained binary-class model, which contributes to producing good optimization results in optimal parameters. The model transfer concept is similar to transfer learning in deep learning [29], where the initial configuration of a model is pre-trained from something else which was trained with similar domains (e.g. pre-training object recognition of a cat prior to transfer learning the model configuration to recognition of a lion). In our methodology, the model transfer is about finding the key model parameters that are influential to the machine learning model performance, from the binary-class classifier to the MCC.

The working logic of this proposed GOSC methodology is shown in two parts in Figures 3a and 3b, respectively. There are two options of operations by the GOSC methodology.In general, the two options share the same initial tasks, such as generating split data subsets according to each of the existing binary classes. Rebalance the majority and minority class data if necessary. Train and optimize each individual binary-class classifier. Record the time performance as well as the model training performance in accuracy and other related performance measures. If an ultimate MCC is not required, perform majority voting, soft or hard, by choice of the user, and obtain the voting result as the final prediction result. The logic is depicted in Figure 3a. If an MCC is required for subsequent predictions, as shown in Figure 3b, a candidate binary-class classifier with the highest prediction accuracy is nominated for model transfer. Its optimized model parameters are copied to the initial configuration of MCC. Thereafter, the MCC is used directly in subsequent prediction without going through a tedious and long parameter optimization process.

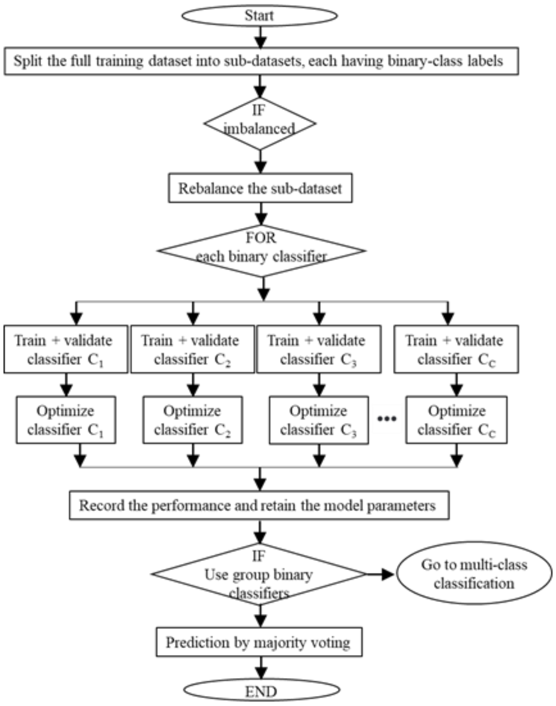
Start
Split the full training dataset into sub-datasets, each having binary-class labels
IF
imbalanced
Rebalance the sub-dataset
FOR
each binary classifier
Train + validate classifier C_1
Train + validate classifier C_2
Train + validate classifier C_3
Train + validate classifier C_C
Optimize classifier C_1
Optimize classifier C_2
Optimize classifier C_3
•••
Optimize classifier C_C
Record the performance and retain the model parameters
IF
Use group binary classifiers
Go to multi-class classification
Prediction by majority voting
END

Fig. 3a. The workflow of Group-of-Single-Class prediction methodology – Part one, when MCC is not needed.

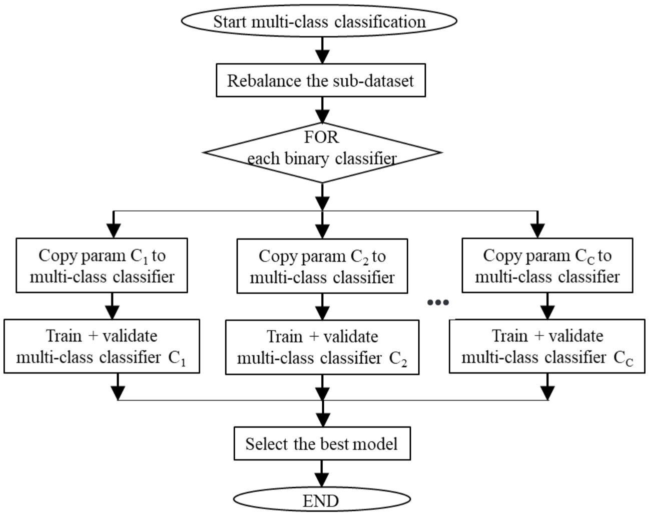
Start multi-class classification
Rebalance the sub-dataset
FOR
each binary classifier
Copy param C_1 to multi-class classifier
Copy param C_2 to multi-class classifier
•••
Copy param C_C to multi-class classifier
Train + validate multi-class classifier C_1
Train + validate multi-class classifier C_2
Train + validate multi-class classifier C_C
Select the best model
END

Fig. 3b. The workflow of Group-of-Single-Class prediction methodology – Part two, when MCC is needed.

IV. Experiment

Two experiments are carried out to validate the concept of GOSC prediction. The first experiment involves using a traditional two-dimensional structured dataset with 70 columns of features that characterize 1000 rows of features. The last column of the dataset is the class categories to which the data map to in classification. There are four class labels in the dataset: normal, carcinoma only, and carcinoma-jaundice. The dataset comes from a physician's donation9 to the HEPAR project [30], run by the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences in collaboration with doctors from the Medical Center for Postgraduate Education. The HEPAR system contains a database of medical records of the gastroenterology clinic of the Warsaw Institute of Nutrition. The data has 70 nominal attributes and 10,000 case histories. The presence or absence of symptoms for 70 signs determines whether a patient suffers from liver disease. An earlier attempt was made to build a causal network based on HEPAR, shown in Figure 4. It could be seen how the attributes have causal relations with one another and with the predicted target as well.

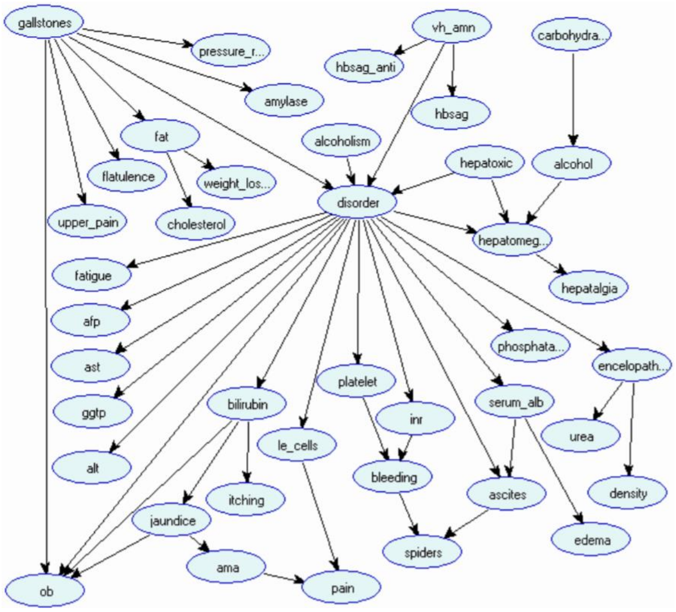

Fig. 4. Belief network of an earlier attempt on HEPAR data

This dataset represents a typical scenario of medical data collection – it started with basic classification between normal and carcinoma. Results of further tests are fused to the data, refining the data with an extra category of carcinoma-jaundice. Of course, these three categories could be extended to more and deeper sub-categories if needed in the future. This illustrates an example of how GOSC could be applicable in boosting the prediction performance when multi-class medical data are dealt with appropriately.

The second dataset came from Kaggle [31]. The dataset carries three classes, each class of images are labeled as normal, infected by bacteria and infected by virus. The images are loaded into a deep learning network powered by Darknet, running on Google Colab GPU environment for training the network. One deep neural network trained as expert SCC for each class of x-ray images. One multi-class network is trained too, which should be able to recognize and distinguish three classes of images during testing the unseen. A total of four deep neural networks are trained; one is for classifying three classes as a multi-class classifier, and three networks as SCCs recognizing only their respective class of images. Figure 5 shows a sample of these three types of x-ray images.

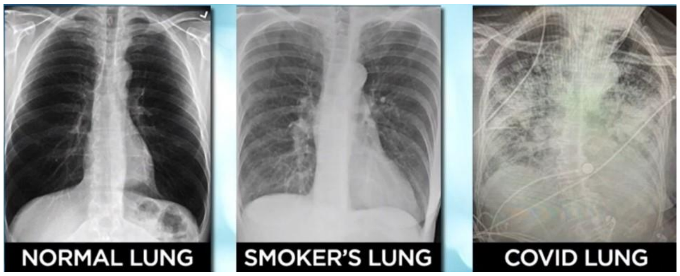

Fig. 5. A sample from each of the three classes of x-ray images

The objective of this experiment is twofold. We need to show that our GOSC works equally well on conventional structured medical datasets using popular machine learning algorithms and x-ray images using convolution-style deep learning, one of the most current medical imaging prediction methods. The other objective is to investigate how the training pattern in the performance curve of training error versus epoch behave in multi-class classification combining the recognition powers of three classes in one model compared to a single expert SCC recognizing an only class of disease ignoring the rest. Intuitively, training a multi-class convolution neural network is more difficult and complex than training a binary-class network. Once this investigation is completed and the hypothesis is established, subsequent studies would be on transfer learning in terms of model parameters transfer (like how GOSC advocates) in sequel experiments.

On the other hand, the HEPAR liver cancer dataset will be used to test GOSC thoroughly. The dataset is first divided into subsets such as a full dataset, the dataset that contains instances of normal and carcinoma-only, and the dataset that contains instances of normal and carcinoma-jaundice. The full dataset naturally and originally consists of instances of the three classes: normal, carcinoma-only, carcinoma-jaundice in the last column of the data matrix. Each data subset constitutes a corresponding model, known as a multi-class model, binary class model 1 (carcinoma-only) and binary class model 2 (carcinoma-jaundice) for short naming.

All three classification models in our experiment will be subject to model optimization. For simplicity, only the two most important model parameters will be optimized to the appropriate values, giving rise to the highest accuracy after optimization. Two-level iterating loops are used to try through the parameter values of the two variables in a wrapper fashion. In each iteration, two candidate values from the testing parameters are used to build a trial model and its accuracy will be measured in three-fold cross-validation. This optimization process is simplest but takes a very long time to loop through combinations of variable values in building, evaluating, and discarding candidate models. This optimization guarantees the best parameter values, thus the best model setup for any given dataset with a given machine learning algorithm. However, in the case of a multi-view classification model, such optimization will take a significantly long time. One of the advantages of GOSC is to eliminate the optimization run on multi-class classification. Instead, finding the right model parameters from its peers – those SCCs from the subsets of the full dataset, and transfer over the values of the parameters from the best performing optimized SCC to the multi-class classifier, without running the optimization from the full dataset for the multi-class classifier.

A collection of representative machine learning algorithms is tested in the experiment. They are two algorithms belonging to tree-based classification – J48 [32], an implementation of Classification and Regression Trees in Java and SPAARC [33], known as a fast decision algorithm. Two algorithms belong to rule-based classification – JRip, which is a propositional rule learner stands for Java Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [34] and a decision list that uses separate-and-conquer (PART) [35]. A classical black-box model by multiple perceptrons [36] is used too. The machine learning algorithms are open-source codes developed by scholars, available on Weka - Waikato Environment for Knowledge Analysis, a machine learning software suite is written in Java developed at the University of Waikato, New Zealand. It is free software licensed under the GNU General Public License.

Five performance evaluation criteria are considered here. They are accuracy, kappa, ROC, FP and time cost. Accuracy is the percentage of correct classifications. Cohen's kappa coefficient (κ) is a statistic that measures the agreement between evaluators on qualitative (categorical) items. Generally, it is considered to be a more robust measure than simple accuracy in data mining, which is simply the number of correctly classified data over the total. Kappa is sometimes taken as a reliability measure for a data mining model. ROC is sometimes known as the AUC-ROC curve in full. AUC (Area Under the Curve) ROC (Receiver Operating Characteristics) curve. It is one of the most important metrics for testing the effectiveness of any classification model. ROC is the probability curve and AUC is the degree or measure of separability. This indicates how well the model can distinguish between classes. The higher the AUC, the better the model predicts that 0 is 0 and 1 is 1. Similarly, the higher the AUC, the better the model can distinguish between diseased and non-diseased patients. The ROC curve is drawn using the ratio of TPR to FPR, where FPR is on the x-axis and TPR is on the y-axis. TPR (True Positive Rate) / Recall /Sensitivity = TP/(TP+FN). Specificity=TN/(TN+FP). FPR=FP/(TN+FP). In general, an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding. FP is FPR as above, known as false alarm rate. The higher it is, the more falsely detected cases as positive, which are negative.. Cost of time is the number of CPU seconds required to create or update a machine learning model in data streams. Accumulated time is the total spent on all training instances. Hardware platform - MacBook Pro with 2.9 GHz Intel Core i5 processor and 8 GB LPDDR3 at 2133 MHz.

The experiment results are charted as bar charts in Figures 6-11, respectively. The machine learning modelIn performance with respect to each of the five indicators (accuracy, kappa, ROC, FP and time cost) for each model induced by each of the five algorithms are compared vis-à-vis over four approaches within the framework of GOSC – (1) original where default model parameters are used without any optimization; (2) full optimization which runs through two-loops-of-iterations for the best pairs of parameters values; (3) model enhancement by taking up the best parameters values that were found from one of the SCCs, and (4) model enhancement by copying over the best parameters values that were found from the other SCC.

Figures 13 and 14 show the deep learning errors in RMSE curves on the logarithmic scale during the model construction process, in terms of current errors and average errors, respectively.

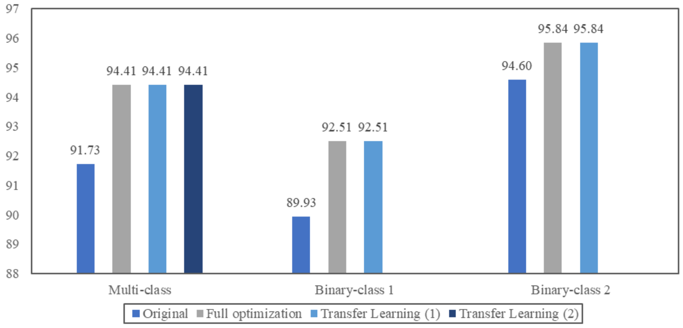

Fig. 6a. Accuracy comparison of models by Decision Tree - J48

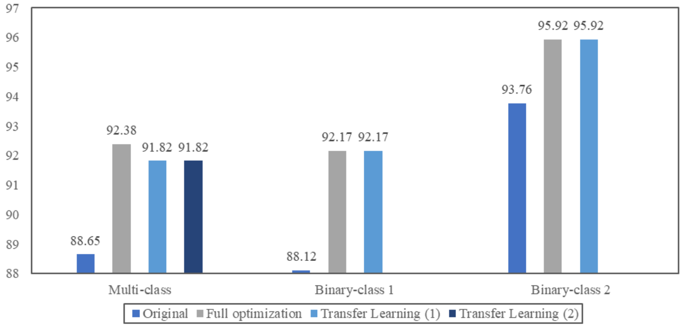

Fig. 6b. Accuracy comparison of models by Decision Tree - SPAARC

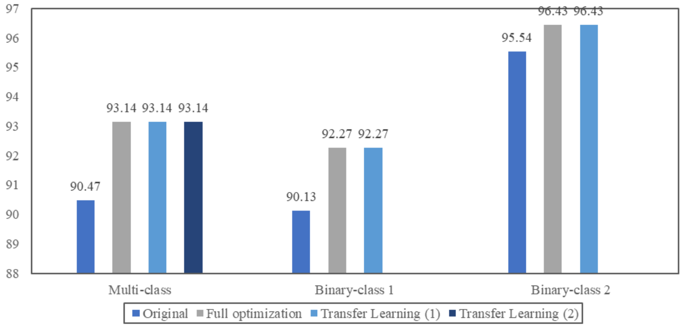

Fig. 6c. Accuracy comparison of models by Rule-based Model - PART

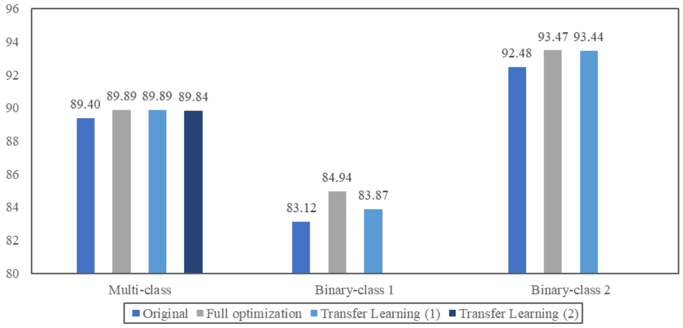

Fig. 6d. Accuracy comparison of models by Rule-based Model - JRip

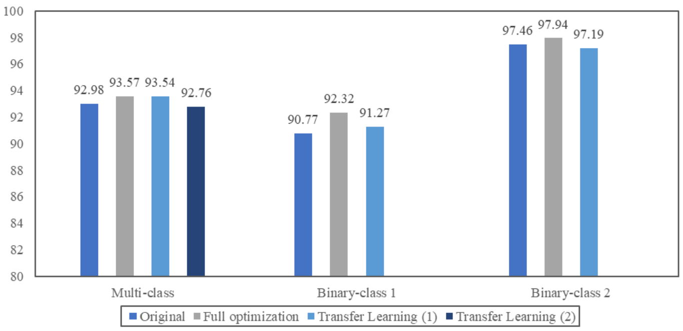

Fig. 6e. Accuracy comparison of models by Black-box Model - Neural Network

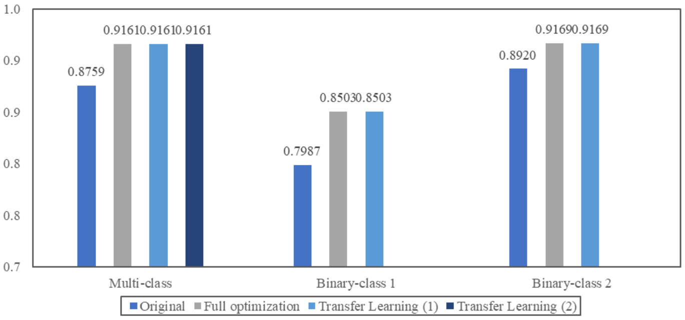

Fig. 7a. Kappa comparison of models by Decision Tree - J48

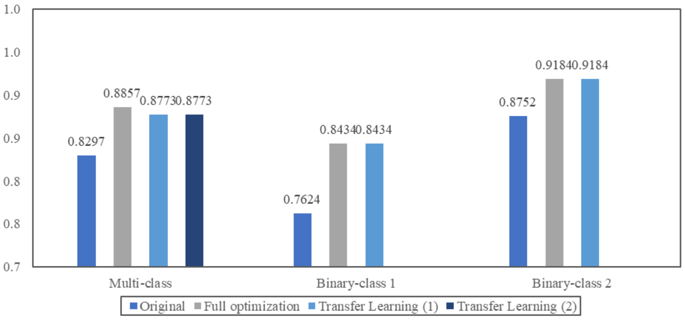

Fig. 7b. Kappa comparison of models by Decision Tree - SPAARC

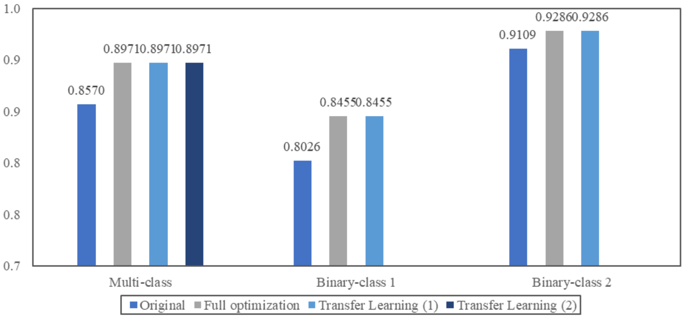

Fig. 7c. Kappa comparison of models by Rule-based Model - PART

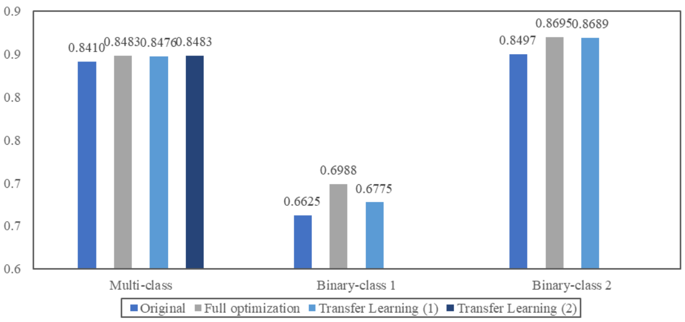

Fig. 7d. Kappa comparison of models by Rule-based Model - JRip

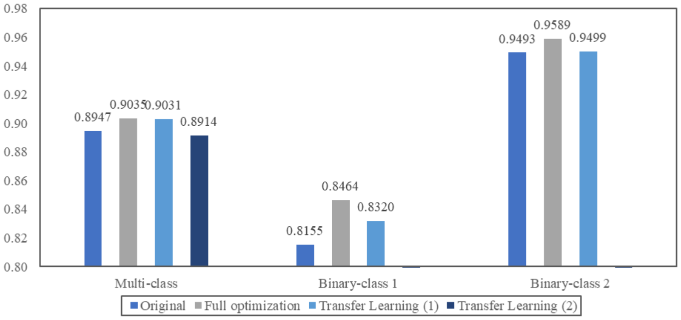

Fig. 7e. Kappa comparison of models by Black-box Model - Neural Network

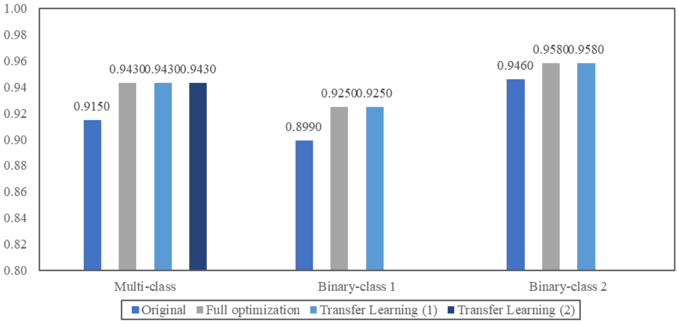

Fig. 8a. ROC comparison of models by Decision Tree - J48

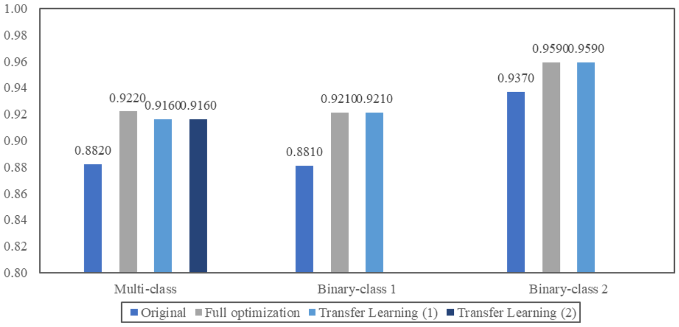

Fig. 8b. ROC comparison of models by Decision Tree - SPAARC

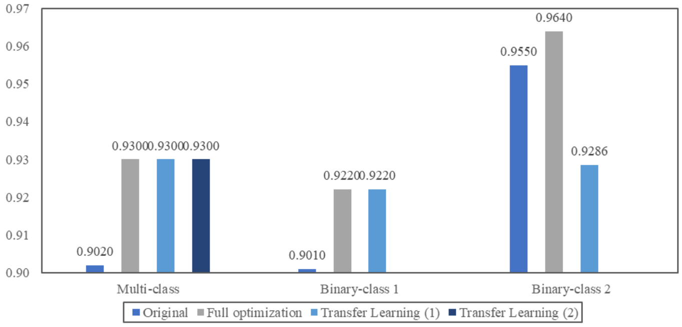

Fig. 8c. ROC comparison of models by Rule-based Model - PART

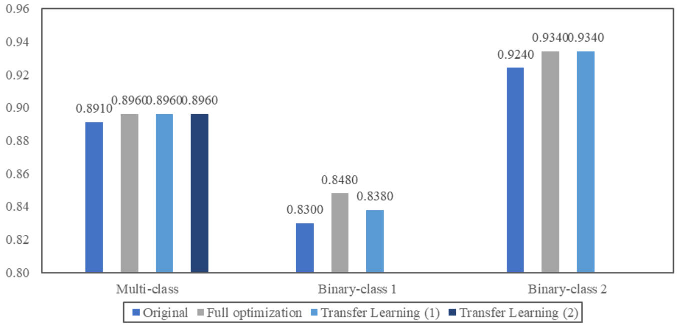

Fig. 8d. ROC comparison of models by Rule-based Model - JRip

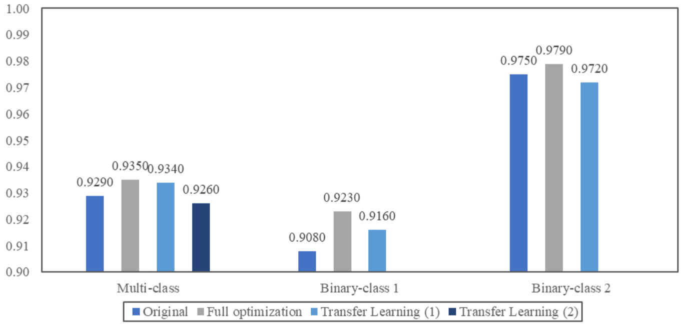

Fig. 8e. ROC comparison of models by Black-box Model - Neural Network

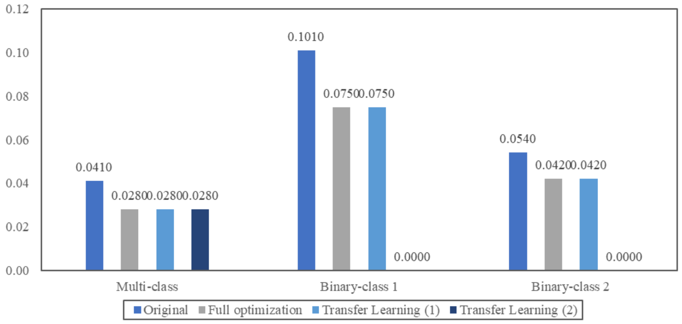

Fig. 9a. FP comparison of models by Decision Tree - J48

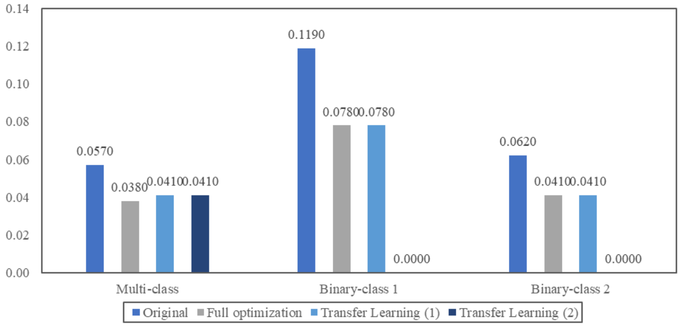

Fig. 9b. FP comparison of models by Decision Tree - SPAARC

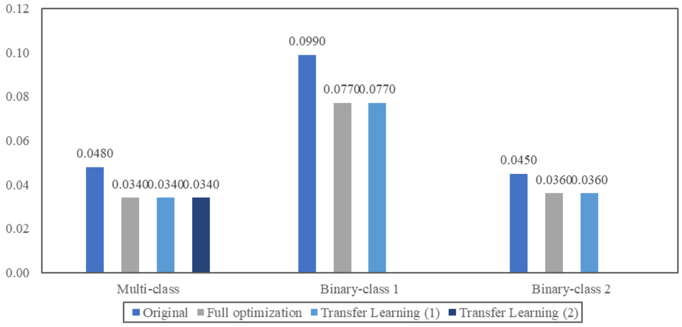

Fig. 9c. FP comparison of models by Rule-based Model - PART

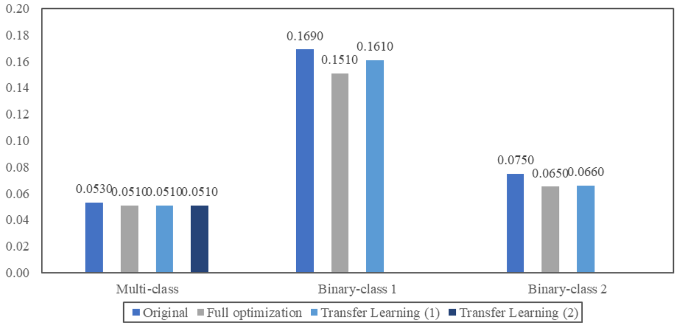

Fig. 9d. FP comparison of models by Rule-based Model - JRip

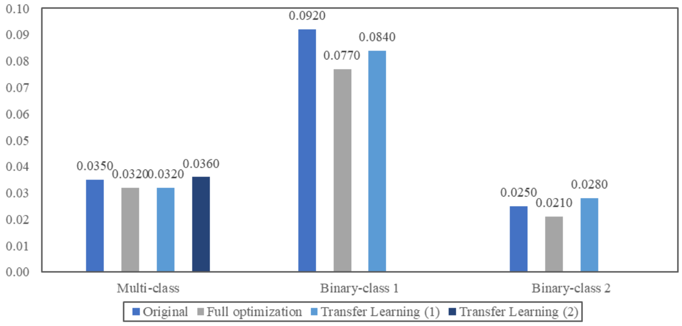

Fig. 9e. FP comparison of models by Black-box Model - Neural Network

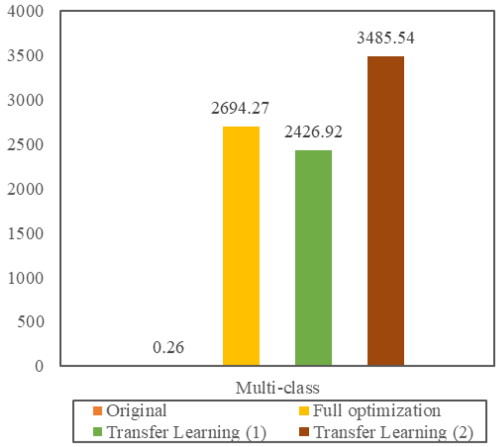

Fig. 10a. Time comparison of models by Decision Tree - J48

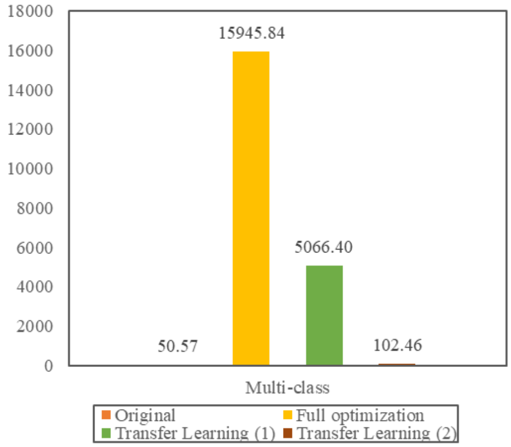

Fig. 10d. Time comparison of models by Rule-based Model - JRip

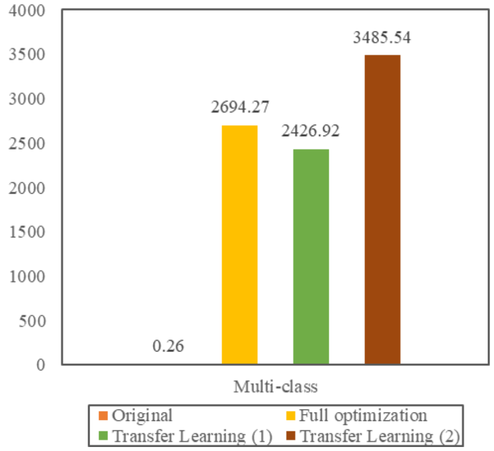

Fig. 10b. Time comparison of models by Decision Tree - SPAARC

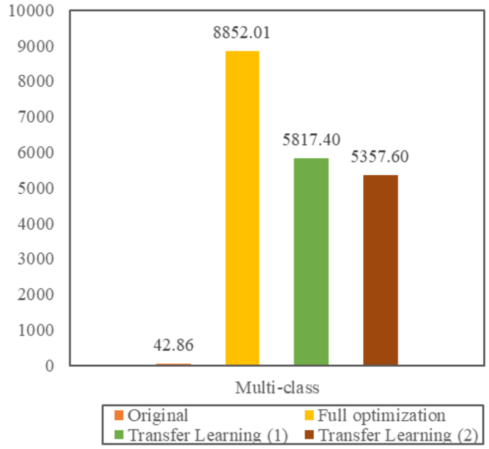

Fig. 10e. Time comparison of models by Black-box Model - Neural Network

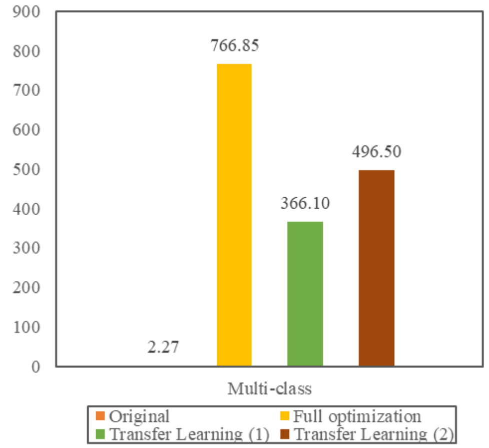

Fig. 10c. Time comparison of models by Rule-based Model - PART

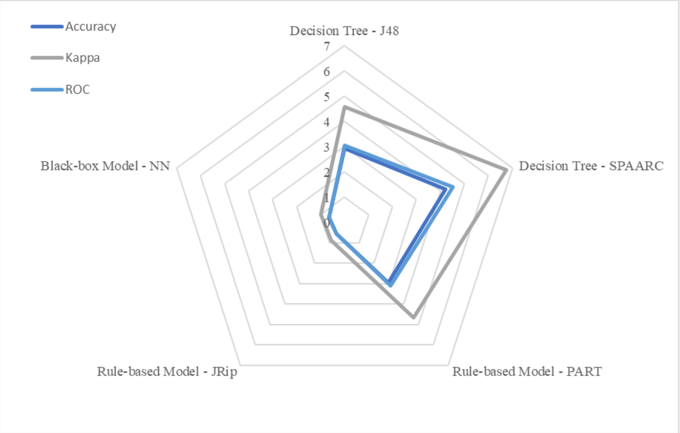

Fig. 11. Radar chart of performance indicators comparison of Accuracy, Kappa and ROC

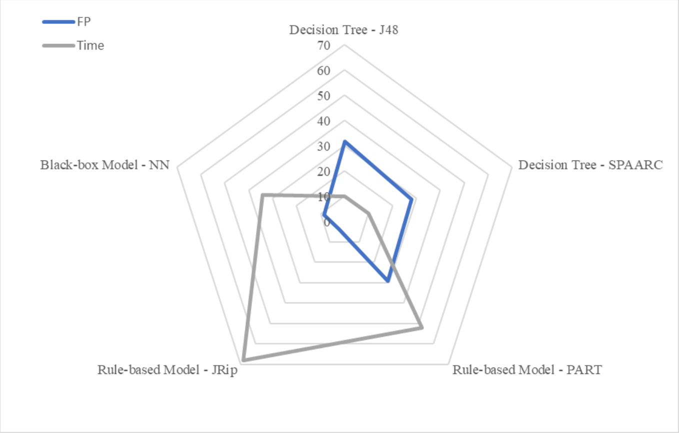

Fig. 12. Radar chart of performance indicators comparison of False Positive Rate and Time cost.

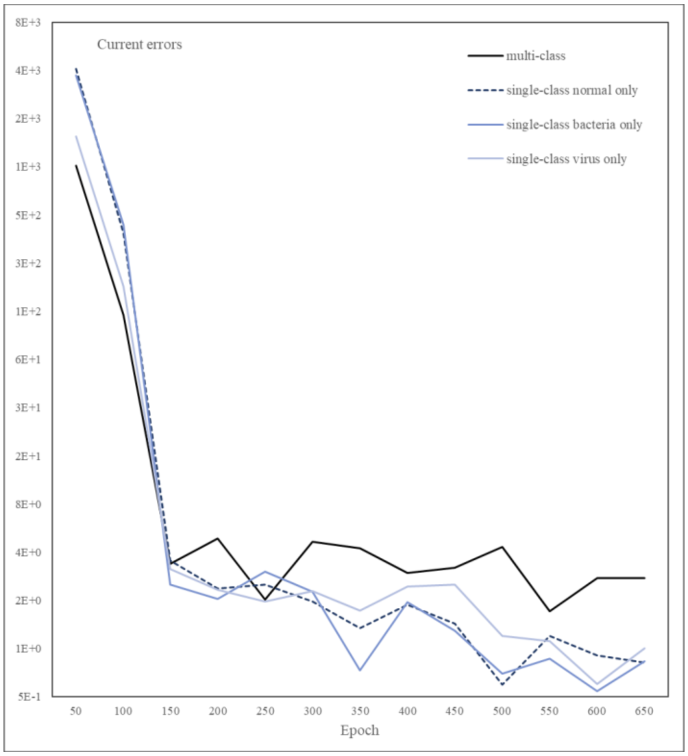

Fig. 13. Current error curves for deep learning models of four types.

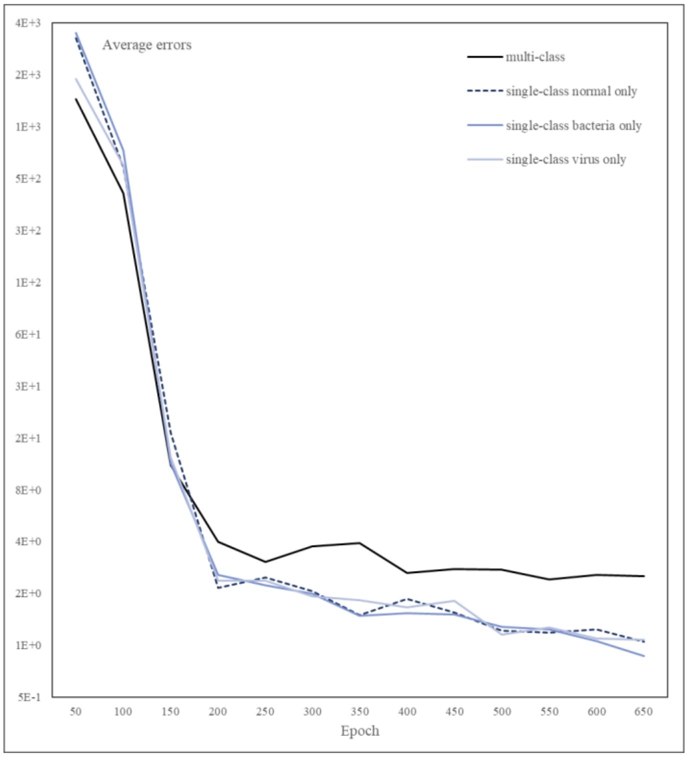

Fig. 14. Average error curves for deep learning models of four types.

V. RESULTS DISCUSSION

Observing over the experiment results from Figures 6-9 clusters of graphs on various performance indicators (accuracy, kappa, ROC and FP), some significant remarks are listed as follow:

1. In general, parameter optimization enhances a model resulting in better performance than the original model in all cases.

2. Full parameter optimization on a multi-class model often generates the best performance, which can be taken as a comparative benchmark.

3. The performance of the multi-class model by any one of the model transfers is close to (and slightly less than), which by full parameter optimization at around 9%.

4. The performances of the multi-class model by the possible model transfers may not always be equal, except for J48 and PART. These two classifiers use information gain as a node selection criterion in common. In an artificial neural network, one of the model transfers is better than the other, different by less than 9%, close to and slightly lower than full optimization by approximately 4.1%.

5. For algorithms PART and SPAARC, there is a very significant improvement using parameter optimization. The performances by full and model transfer optimizations are very close too. Therefore, it suggests that model transfer is quite a feasible solution to trim down the optimization time while significantly enhancing the prediction performance.

6. By the designs of the algorithms, JRip and artificial neural networks are relatively strong and stable models. However, optimizing the parameters helps marginally increase the performance compared to the other algorithms under test. In other words, these two algorithms do not show much

improvement when their model parameters are optimized. For example, in Figure 9d, the false-positive rates for JRip are very similar between optimized and otherwise models.

7. For binary-class models, copying the parameters that have been optimized from another binary-class model produce similar but less perfect performance compared to its own optimization. But it still outperforms the original model without optimization at any rate.

As a concluding remark, a full parameter optimization yields the best performance; however, it could be replaced by the model transfer method as there shows little difference between full and model transfer optimizations.

While known as time cost, the timing performance for each experiment run is measured from the beginning of model construction to the end. That excludes multi-fold evaluation time. It can be seen from Figures 10a-e where that full optimization is always the most time-consuming approach. The difference between a model being optimized and original could be up to 100 folds, as in Figure 10d for JRip. On the bright side, the time taken for model construction by using model transfer is always much lower than that for full optimization. This proves that it is possible to use model transfer in lieu of full optimization to achieve on-par performance at only a fraction of the time cost.

In Figures 11 and 12, the five performance indicators are stretched over radar charts over the five testing algorithms in vertices. The results are shown in marginal percentage gains with respect to the performance improvement using model transfer optimization over the original model without any optimization. In Figure 11, it can be clearly observed that algorithms such as artificial neural networks and JRip only have marginal performance increases in control of the other three algorithms. Decision tree SPPAARC has the greatest gain when model transfer optimization is used, in all the accuracy, kappa and ROC. The runner-up algorithms, J48 and PART, show the same. In particular, Kappa has gained the most compared to Accuracy and ROC using optimization. That means the models have become more reliable, being able to generalize well. In Figure 12, the results of false-positive rate and time costs are laid over the radar chart of five algorithms. Again, the two stable algorithms, artificial neural networks and JRip have little performance gains in false-positive rate. J48, SPAARC and PART, which are largely decision tree-based algorithms, greatly reduced false-positive rates. Ironically, JRip has the greatest gain in reducing the time up to almost 700% when it comes to time cost-saving. Artificial network networks and PART also have a significant reduction in time cost when model transfer optimization is used. In contrast, J48 and SPAARC have already been quite fast in model construction, with little time cost gain and optimization. In summary, false-positive rates are lowered by approximately 300% for tree-based algorithms, and time cost is hugely saved for JRip as well as neural network and PART when transfer model optimization is used.

GOSC is also tested on deep learning models of several types. According to the given COVID19 lung infection dataset, four deep learning model trainings were set up. Each model is trained with a particular dataset of various classes. The single class models are trained with datasets with only individual classes, i.e., single-class normal, single-class bacteria, and single-class virus. The multi-class model is trained with the dataset containing all three classes - normal, bacteria and virus.

In deep learning, it is anticipated that error curves descend sharply in the early period of epochs. Then the curves decay to equilibrium as the errors continue to drop at a decreasing rate. In Figure 13, it is observed that the error curve of the multi-class model has the least error relative to the single-class models at the early descend. However, the curve of the multi-class model remains higher than the rest of the error curves of the other single-class models. This observation indicates two phenomena: the model is learned better when multi-class training data are availablethan to single-class monotonous training data. Secondly, the multi-class model finds it hard to converge to a very low steady-state error rate in the period of curves decay. In contrast, the single-class models decay sooner than the multi-class model and have lower error rates than the multi-class model. It is due to the complexity of the multi-class learning and the data. The latter phenomenon essentially hints that the operation involved in multi-class learning takes takes longer than single-class learning. The complexity infers that a full optimization (which is organic to multiple executions of learning in search of the right parameters) at a multi-class model will be at high costs, in terms of time and difficulty to attend a reasonable low error rate. On the other hand, a single-class model can achieve it relatively more easily. This implies it is potentially possible and feasible to use model transfer copy under GOSC methodology to enhance the multi-class model and save time on model parameter optimization. It is noted that GOSC methodology would not only be applicable in medical domain. Other domains such as speech recognition, computer network load balancing, and remote health monitoring, where supervised learning is the most focused, would be benefited by GOSC.

VI. Conclusion

A novel computing methodology called Group-of-Single-Class prediction (GOSC) coupled with majority voting and model parameter transfer is presented in this paper. GOSC is for attaining optimally high (or near best) accuracy for multi-class classification, using the model transfer method from the binary-class model, which is lighter and much quicker than full optimization. The binary-class model is built on the same training dataset as the multi-class model. Two sets of experiments were conducted, one on a structured two-dimensional data matrix with instances about patients who might have liver carcinoma and other complications. The other dataset is a collection of X-ray images of three groups of normal patients, who are suffering from lung pneumonia infected by bacteria, or who are infected by virus. The datasets are made into multi-class and binary-class compatible for experiments. Having both data types representing electronic medical records in the two most popular formats in our experimentation, GOSC was tested, and the results were satisfactory. Decision tree types of classifiers gained up to 4.2% for SPAARC and 2.9% for J48 and PART. Their false positive rates are primarily reduced by31.7% for J48 and around 28% for others. Kappa statistics could be interpreted as generalization ability upon testing unseen data. Generally, there is improvement using GOSC from 4.6% to 6.7% except for JRip and artificial neural networks that hardly can reach up to 1%. They are quite stable with little improvement in accuracy in kappa. However, they gain the most from time-cost. While J48 and SPAARC, which are the tree-type classifiers, gained 9.9% in saving time in model optimization. In contrast, JRip and artificial neural networks gained as high as 68.2% and 34.3%, respectively. Overall, GOSC via simulation experimentation is shown to achieve an enhanced prediction performance to an almost generally, aximum extend by using the model transfer method instead of full optimization. In all cases, GOSC shows its advantages in terms of maximizing the performance without a very high time cost. As future works, more medical records are to be tested, and full transfer learning, including hyperparameter optimization with GOSC for convolution neural networks, is to be implemented and tested.

References

[1] Dhruva, S.S., Ross, J.S., Akar, JG et al. "Aggregating multiple real-world data sources using a patient-centered health-data-sharing platform". npj Digit. Med. 3, 60 (2020).

[2] Uddin, S., Khan, A., Hossain, M. et al. "Comparing different supervised machine learning algorithms for disease prediction". BMC Med Inform Decis Mak 19, 281 (2019).

[3] Runkang Ding, Fan Jiang, Jingui Xie & Yugang Yu (2017) "Algorithmic prediction of individual diseases", International Journal of Production Research, 55:3, 750-768, DOI: 10.1080/00207543.2016.1208372

[4] T. Nguyen et al., "PAN: Personalized Annotation-based Networks for the Prediction of Breast Cancer Relapse," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, doi: 10.1109/TCBB.2021.3076422

[5] Bayati M, Bhaskar S, Montanari A. "A Low-Cost Method for Multiple Disease Prediction". AMIA Annu Symp Proc. 2015 Nov 5;2015:329-38. PMID: 26958164; PMCID: PMC4765607.

[6] Runzhi Li, Wei Liu, Yusong Lin, Hongling Zhao, Chaoyang Zhang, "An Ensemble Multilabel Classification for Disease Risk Prediction", Journal of Healthcare Engineering, vol. 2017, Article ID 8051673, 10 pages, 2017. https://doi.org/10.1155/2017/8051673

[7] D.-Y. Yeh, C.-H. Cheng, and Y.-W. Chen, "A predictive model for cerebrovascular disease using data mining," Expert Systems with Applications, vol. 38, no. 7, pp. 8970–8977, 2011.

[8] B. L. Shivakumar and S. Alby, "A survey on data-mining technologies for prediction and diagnosis of diabetes," in 2014 International Conference on Intelligent Computing Applications, pp. 167–173, Coimbatore, 2014.

[9] H. Neuvirth, M. Ozery-Flato, J. Hu et al., "Toward personalized care management of patients at risk: the diabetes case study," in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 395–403, San Diego, CA, USA, August 2011.

[10] S. Sankaranarayanan and T. P. Perumal, "A predictive approach for diabetes mellitus disease through data mining technologies," in 2014 World Congress on Computing and Communication Technologies, pp. 231–233, Trichirappalli, 2014.

[11] Allwein, E., Schapire, R., & Singer, Y. (2000). "Reducing multi-class to binary: a unifying approach for margin classifiers". Journal of Machine Learning Research, 1, 113--141

[12] Dong L., Frank E., Kramer S. (2005) "Ensembles of Balanced Nested Dichotomies for Multi-class Problems". In: Jorge A.M., Torgo L., Brazdil P., Camacho R., Gama J. (eds) Knowledge Discovery in Databases: PKDD 2005. PKDD 2005. Lecture Notes in Computer Science, vol 3721. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11564126_13

[13] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, Francisco Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes", Pattern Recognition, Volume 44, Issue 8, 2011, Pages 1761-1776, ISSN 0031-3203

[14] J. Fürnkranz, "Round robin ensembles", Intelligent Data Analysis, 7 (5) (2003), pp. 385-403

[15] Salciccioli J.D., Crutain Y., Komorowski M., Marshall D.C. (2016) "Sensitivity Analysis and Model Validation. In: Secondary Analysis of Electronic Health Records". Springer, Cham. https://doi.org/10.1007/978-3-319-43742-2_17

[16] S. Duan, Y. Li, Y. Wan, P. Wang, Z. Wang and N. Li, "Sensitivity Analysis and Classification Algorithms Comparison for Underground Target Detection," in IEEE Access, vol. 7, pp. 116227-116246, 2019, doi: 10.1109/ACCESS.2019.2936132.

[17] Li J, Fong S, Sung Y, Cho K, Wong R, Wong KKL. "Adaptive swarm cluster-based dynamic multi-objective synthetic minority oversampling technique algorithm for tackling binary imbalanced datasets in biomedical data classification". BioData Min. 2016;9:37. Published 2016 Dec 1. doi:10.1186/s13040-016-0117-1

[18] R. Kohavi (1995). "Wrappers for Performance Enhancement and Oblivious Decision Graphs". Department of Computer Science, Stanford University.

[19] Kotthoff L., Thornton C., Hoos H.H., Hutter F., Leyton-Brown K. (2019) "Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA". In: Hutter F., Kotthoff L., Vanschoren J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_4

[20] Bergstra, J., Bengio, Y.: "Random search for hyper-parameter optimization". JMLR 13, 281–305 (2012)

[21] Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: "Algorithms for Hyper-Parameter Optimization". In: Proc. of NIPS-11 (2011)

[22] Yang Zhongguo, Li Hongqi, Sikandar Ali and Ao Yile, "Choosing Classification Algorithms and Its Optimum Parameters based on Data Set Characteristics", Journal of Computers Vol. 28, No. 5, 2017, pp. 26-38

[23] A. Narasimhamurthy, "Theoretical bounds of majority voting performance for a binary classification problem," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 12, pp. 1988-1995, Dec. 2005, doi: 10.1109/TPAMI.2005.249.

[24] A. Dogan and D. Birant, "A Weighted Majority Voting Ensemble Approach for Classification," 2019 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 1-6, doi: 10.1109/UBMK.2019.8907028.

[25] An K., Meng J. (2010) "Voting-Averaged Combination Method for Regressor Ensemble". In: Huang DS., Zhao Z., Bevilacqua V., Figueroa J.C. (eds) Advanced Intelligent Computing Theories and Applications. ICIC 2010. Lecture Notes in Computer Science, vol 6215. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14922-1_67

[26] Michael Beyeler, "Understanding different voting schemes", Oreilly, [Last Accessed on 14 May 2021], https://www.oreilly.com/library/view/machine-learning-for/9781783980284/47c32d8b-7b01-4696-8043-3f8472e3a447.xhtml

[27] Ana Carolina Lorena, André CPLF de Carvalho, "Evolutionary tuning of SVM parameter values in multi-class problems", Neurocomputing, Volume 71, Issues 16–18, 2008, Pages 3326-3334, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2008.01.031.

[28] Tharwat, A., Gabel, T. Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. Neural Comput & Applic 32, 6925–6938 (2020). https://doi.org/10.1007/s00521-019-04159-z

[29] Ho, N., Kim, YC. Evaluation of transfer learning in deep convolutional neural network models for cardiac short axis slice classification. Sci Rep 11, 1839 (2021). https://doi.org/10.1038/s41598-021-81525-9

[30] Agnieszka Onisko, Marek J.Druzdzel,and Hanna Wasyluk, "A Bayesian Network Model for Diagnosis of Liver Disorders", In Proceedings of the Eleventh Conference on Biocybernetics and Biomedical Engineering, pages842-846, Warsaw, Poland, December2-4, 1999

[31] Tawsifur Rahman et al., "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images", Computers in Biology and Medicine, Volume 132, 2021, 104319, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2021.104319.

[32] Ranjit Panigrahi, Samarjeet Borah, "Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets", Procedia Computer Science, Volume 132, 2018, Pages 323-332, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.186.

[33] Yates D., Islam M.Z., Gao J. (2019) "SPAARC: A Fast Decision Tree Algorithm". In: Islam R. et al. (eds) Data Mining. AusDM 2018. Communications in Computer and Information Science, vol 996. Springer, Singapore. https://doi.org/10.1007/978-981-13-6661-1_4

[34] William W Cohen, "Fast Eective Rule Induction", Proceedings of the Twelfth International Conference of Machine Learning, pp.1-10. 1995.

[35] Eibe Frank, Ian H. Witten: "Generating Accurate Rule Sets Without Global Optimization". In: Fifteenth International Conference on Machine Learning, 144-151, 1998.

[36] Bikku, T. Multi-layered deep learning perceptron approach for health risk prediction. J Big Data 7, 50 (2020). https://doi.org/10.1186/s40537-020-00316-7

[37] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto and A. Sciarrone, "Smart and Robust Speaker Recognition for Context-Aware In-Vehicle Applications," in IEEE Transactions on Vehicular Technology, vol. 67, no. 9, pp. 8808-8821, Sept. 2018, doi: 10.1109/TVT.2018.2849577.

[38] J. Liu, Y. Kawamoto, H. Nishiyama, N. Kato and N. Kadowaki, "Device-to-device communications achieve efficient load balancing in LTE-advanced networks," in IEEE Wireless Communications, vol. 21, no. 2, pp. 57-65, April 2014, doi: 10.1109/MWC.2014.6812292

[39] I. Bisio, F. Lavagetto, M. Márchese and A. Sciarrone, "Comparison of situation awareness algorithms for remote health monitoring with smartphones," 2014 IEEE Global Communications Conference, 2014, pp. 2454-2459, doi: 10.1109/GLOCOM.2014.7037176