Mathematical Problems of Computer Science 60, 40–51, 2023.
doi: 10.51408/1963-0108

UDC 004.8

Multi-Stage Classification Scheme to Optimize Medical Treatments

Karen M. Gishyan

Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
e-mail: karengishyan.res@outlook.com

Abstract

In most existing machine learning and deep learning settings, classification and regression prediction problems may be described as a process where the model output is based on a single-stage input. In most real-life scenarios achieving the desired medical state for the patient may involve dynamically solving drug prescription problems based on the input data at different stages, where each stage is a logical grouping such as timestep division, ICU stay, etc. Data at a given stage represents a recovery progression and can be fundamentally different from the datasets from the previous and future stages. Although A single model may solve the task, a multi-stage learning procedure may be more suitable. To solve this task, we propose an FNN-driven ensemble-based approach for predicting the medications that the patient should receive at each stage of the recovery process. The final medical discharge location is predicted as a result of sequential predictions of drugs and features. In this work, we combine model ensembling and multi-stage iterative learning for solving an optimal drug prescription generation task as a contribution to the existing literature.
Keywords: Multi-stage classification, Treatment-optimization, Model-ensembling, Machine learning.
Article info: Received 21 March 2023; sent for review 2 May 2023; received in revised form 3 July 2023; accepted 4 September 2023.
Acknowledgement: The author would like to thank some of the department members for their valuable feedback.

1. Introduction

In this work, we propose a feedforward neural network-based iterative ensemble model architecture for solving the target class classification task for the medical domain, where the target class is the *Home Discharge* [1]. The proposed approach involves *Feature*, *Drug*, and *Output* prediction networks, which can be considered as an extension of decomposition-based (e.g., divide and conquer) and multi-objective optimization-based ensemble methods. Similar to

[1]Code source: `https://github.com/karen-gishyan/project`

some of decomposition methods, the original dataset is divided into a collection of datasets for multiple sub-processing, and similar to multi-objective optimization, we output multiple diverse predictors instead of a single predictor. Unlike decomposition methods, single *Drug* and *Feature* networks are optimized on different datasets at different stages instead of multiple predictors being optimized of the same type. This approach provides flexibility in adding more stages to the model architecture. This also means that the same set of network parameters is updated multiple times during a single forward pass. The alternative could be to construct a different network for each of the input datasets for each stage, where the downside would be that the number of networks that need to be optimized linearly would increase with each additional stage in the model architecture. Training is divided into three fixed stages (steps). For each stage s except the last one, we train two distinct FNN models: the *Drug* prediction network and the *Feature* prediction network, which predict drugs and features for the next stage $s + 1$, respectively. The feature and drug prediction training inputs at stage $s + 1$ are concatenated with the features and drugs at stage s. The rationale behind this approach is that the drugs given to a patient at each stage depend not only on the drugs at the previous stage but also on the patient's features at the previous stage. The same logic applies to the features, which depend not only on the past features but also on the past medication. We believe this approach results in more diverse datasets and better learning. Currently, the actual datasets for model training are available only at the first stage, and the training datasets at each stage s are iteratively generated based on the predicted features and drugs using only the observed data at the first stage as an initialization point. For the last stage, we train an *Output* model for learning the discharge location classes. Overall, we use three models across all stages: *Drug* network, *Feature* Network, and *Output* network. The main method can be considered a model-based approach with a strong emphasis on data-driven stage-based logic, data components, and logical variations of which were first described in [1] and [2] and further detailed in this work. The contributions can be summarized in the following points:

- We provide an ensemble-based iterative classification/regression pipeline that includes not just one but three different networks, each being optimized simultaneously in the forward pass. In addition, each *Feature* and *Drug* network is used multiple times depending on the number of stages. This is different from some of the existing approaches, where a new network is initialized at each stage.

- Having patient features for time t, we predict treatment for $t + n$ periods, where n is the number of stages (3 for this experiment).

- We train our models on synthetic datasets and give a detailed description of the stage-based data preprocessing technique.

2. Related Work

Ensemble methods have been widely used in research fields such as computational intelligence and machine learning. Ensemble methods can be categorized into conventional ensemble methods such as bagging, boosting and random forest, decomposition, negative correlation learning multi-objective optimization-based methods, fuzzy ensemble, multiple kernel learning ensemble, and deep learning ensemble Diversity is important in ensemble methods, and the three ways to create diversity are data diversity, parameter diversity,

and structural diversity [3, 4, 5]. Data diversity creates multiple datasets from the input dataset to train different models. The more diverse the datasets, the more diverse the model learning. Parameter diversity uses different parameter sets for generating different base predictors, and even with the same training set, the output of the predictors may differ. In structural diversity, ensemble predictors have different structures and architectures, and this kind of ensemble is also known as a heterogeneous ensemble [3]. Besides data, parameter, and structural diversity methods, there are other methods such as divide and conquer [6], multi-objective optimization [7], and fuzzy ensemble. In multi-objective classification, the training process yields a collection of optimal and diverse predictors instead of a single predictor [3]. Divide and conquer is mostly seen in time series forecasting, where the original dataset is often divided into a collection of parallel or hierarchical datasets, forming subtasks. Predictors are applied to each subtask, and then the outputs are aggregated. Datasets usually have different characteristics, and predictors mainly differ from each other. In divide and conquer methods, the original time series is decomposed into a collection of time series from which the original series can be reconstructed [6]. The goal is to obtain smaller and simpler time series, apply predictions to the decomposed time series, and aggregate the predictions. Both seasonal decomposition and wavelet transform are decomposition-based ensemble methods. While the first approach implies a prediction algorithm such as an SVR being applied to each seasonally decomposed component, in the second approach, a prediction algorithm is applied to the sub-series obtained by decomposing the original series into orthonormal series by the time domain [6]. [8] presents a divide-and-conquer-based hierarchical optimization framework for ensemble classifier learning. The framework includes a data training environment (DTE) creation that divides the data into multiple clusters and then trains heterogeneous base classifiers, which are later combined for an optimal ensemble. For optimizing multi-stage cascade classifiers, [9] proposes a deep model, which jointly optimizes multiple classifiers through several stages of backpropagation. Cascade classifiers were first proposed in [10] for solving a multi-stage recognition problem. Since then, cascading classifiers have been successfully applied to tasks such as image recognition [11], name entity recognition in clinical notes [12], anomaly detection and localization [13], and so on.

There are a few examples of multi-stage classification used in the medical domain. [14] solves a multi- stage classification problem for HER2 breast cancer by proposing a transfer learning-based approach used on the BCI dataset. [15] proposes an effective feature ensemble with multi-stage classification for breast cancer diagnosis, and the verification happens on a publicly available mammogram image dataset collected from the IRMA project. [16] proposes an automatic system involving multi-stage classification for diagnosing congestive heart failure using short-term heart rate variability analysis. For the experiments, open databases from Physionet, Normal Sinus Rhythm Database (NSR2DB), and Congestive Heart Failure Rhythm Database (CHF2DB) are used. [17] uses a multi-stage approach for performing arrhythmia recognition and classification. [18] uses a machine learning-based multi-stage classification method to classify Alzheimer's disease more efficiently. [19] uses a two-stage machine learning classification approach for heart disease prediction. [20] proposes a two-stage multi-modal learning algorithm for multi-label skin disease classification. [21] proposes a multi-stage approach to detect tumors, classify them into glioma or meningioma and perform their segmentation. [22] uses a transformer-based model for automatic multi-stage classification of diabetic retinopathy. [23] uses multi-stage superpixel classification for classifying four lung diseases and healthy lungs using chest X-ray images.

3. Data

For the experiments from the *MIMIC-III* clinical database, we use *ADMISSIONS*, *D_Items*, *PRESCRIPTIONS*, datasets, and a subset of the *CHARTEVETS* dataset, the latter containing 5 million rows from the whole dataset. We use the datasets listed above for generating *Features* and *Drugs* datasets. The *Features* dataset includes admissions and a list of those features for which the patient has had between 10 and 300 measurements throughout the stay. Similar logic is applied to the *Drugs* dataset, where we select the admission for which the patient was given more than 10 drugs. These two datasets are further filtered based on the admission IDs that are present in both. This approach is performed for the *DIABETIC KETOACIDOSIS* diagnosis.

3.1 Stage-Based Features and Drugs

As a result of data processing, where we generate the initial versions of the *Features* and *Drugs* datasets, the results of which can be observed in Table 1, we proceed to do extra stage-based processing to obtain the datasets for each time-stage.

3.1.1 Step 1 Processing

From *Features* and *Drugs*, we filter those observations where the patient's stay length was between 6 and 8 days. We define three stages and generate one *Features* dataset, and one *Drugs* dataset for each stage and one *Output* dataset only for the third stage. The first stage is defined as the *Initial* stage, the second as the *Intermediary* stage, and the third stage as the *Final* stage. The features are *O2 saturation pulse oximetry*, *Heart Rate*, *Respiratory Rate*, *Non-Invasive Blood Pressure mean*, *Non-Invasive Blood Pressure systolic*, *Non-Invasive Blood Pressure diastolic*, *Temperature Fahrenheit*, *Arterial Blood Pressure systolic*, *Arterial Blood Pressure diastolic*, *Arterial Blood Pressure mean*. We define a *stage* to be a period corresponding to 2 days spent at the hospital for the first two stages and between 2 and 4 days spent at the hospital for the final third stage. This means that the features observed and drugs given for the first 2 days become the *Features* and *Drugs* datasets for the first stage, the ones for the 3rd and 4th days the datasets for the second stage, and from the 5th day up to the 6th or to the 8th day, depending on admission, the datasets for the third stage. The rationale behind choosing these numbers for defining the stages is that we want each stage to have at least 2 days' worth of data. The final stage can have up to two days of more data compared to previous stages to loosen the restriction for the total admission duration to be precisely 6 days. However, a bigger difference in the number of days between stages means more time series observations for some stages compared to others, and we keep it to 2 to avoid increasing the tradeoff further. The more the difference between observations, the more value padding should be performed to make sure that the datasets have the same shape.

At this point, we have 3-dimensional data, where the first dimension (batch) is the given admission, the second dimension (rows) is the number of time steps for each of the given stage, and the third dimension (columns) is the features or the drugs, depending on if the dataset is the *Features* or the *Drugs*. The structure of the *Features* dataset can be seen in Table 2.

Table 1. Initial preprocessing results.

Dataset Statistics		
	Features Dataset	**Drugs Dataset**
Original		
N Observations	2949897	4945985
Unique Admissions	3869	47031
Unique Diagnoses	1445	13880
After Filtering		
N Observations	2948538	474213
Unique Admissions	3859	3859
Unique Diagnoses	1443	1443

Table 2. *Features* dataset for a single batch before averaging.

Features Dataset				
Timestep	**Feature 1**	**Feature 2**	$\cdots$	**Feature 10**
t_1	value 1	value 1	$\cdots$	value 1
t_2	value 2	value 2	$\cdots$	value 2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_{10}	value 10	value 10	$\cdots$	value 10

3.1.2 Step 2 Processing

Processing of this stage allows us to obtain datasets that will be used as a basis of synthetic data generation used for modeling and experimentation. The processing described in Section (3.1) has one limitation. Each patient will surely have a different number of drugs given and a different number of charted feature measurements for each stage. As most deep learning frameworks, including the one used in this work, assume that each batch input (admission data) for the model has the same shape, this means that all the batches need to be padded with a predefined value for the input data to have a certain shape of (i, j, k). After padding, we remove the second dimension by averaging the time series instances over the rows, which we believe makes the data for each stage more representative and less dependent on a single time-stage observation, which in most cases would be an artificially padded value. For the *Drugs* dataset, each drug is first encoded and is assigned a discrete numerical value, but as there is no natural ordering between the drugs, we further do a dummy conversion, which means each drug will be present for each patient in the form of either 0 or 1 (absent or present) showing whether the given drug was part of the patient's treatment procedure for the given stage. At this stage, we also generate an *Output* dataset, which again holds values of 0 or 1, which are later used for the binary classification task. To summarize, we end up with three datasets. In stages 1, 2, and 3, we get *Features*, *Drugs*, and *Output* datasets with

the following shapes $(n, 10)$, $(n, 902)$, and $(n, 1)$, respectively, where n is the number of rows or unique admissions. Note that the *Output* dataset is only present for the 3rd stage.

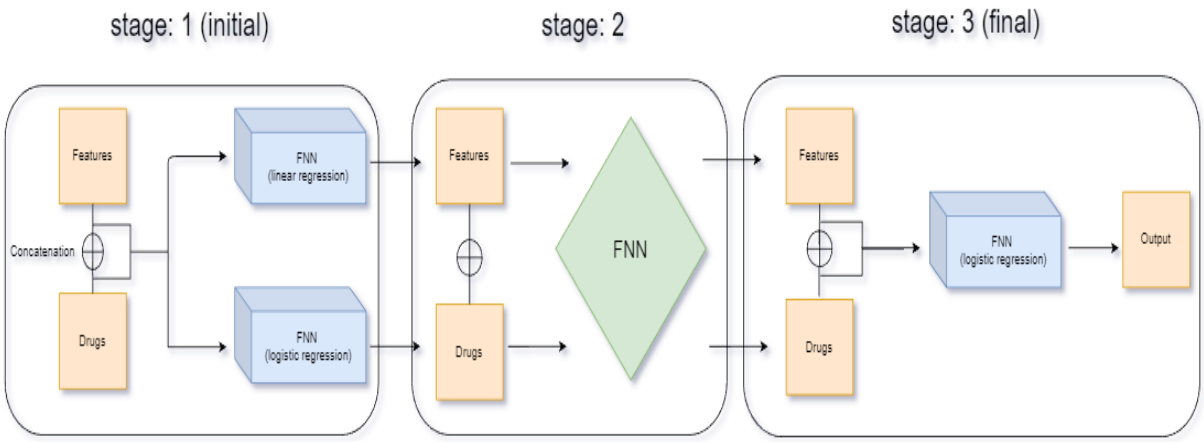

Fig. 1. Multi-stage classification pipeline.

4. Methodology

4.1 Synthetic Data Generation

Due to the particular data processing logic, the *Drugs* and *Features* datasets that we obtained for *DIABETIC KETOACIDOSIS* have only 16 observations, a shape of $(16, 902)$. To be able to train a machine learning model, we augment our datasets using a synthetic data generation technique using the Synthetic Data Vault (SDV) package [24]. We use a TVAE model, a VAE-based deep learning data synthesizer [25]. We augment each dataset to a size of 5000 observations; however, not all the observations are eventually used for model fitting. The average synthesized *Features* dataset similarity to the original *Features* dataset across three stages is 64%, while the synthesized *Drugs* dataset similarity to the original *Drugs* dataset across three stages is 98%. Output is generated as part of the *Features* for the third stage. As you can see, the similarity is very high for the *Drugs* dataset, which is understandable as the data is in a binary format compared to the continuous *Features* dataset. There is still, however, a significant class imbalance in the augmented datasets. For demonstration, there are only 184 instances where the final discharge outcome was positive (labeled as 1), while for the rest of the observations, the labeling is 0. The model training suffers from such imbalance, and the training results are poor. To overcome this problem, from dataset instances that have discharge outcomes of 0, we filter the first 184 instances and concatenate them to the other 184 instances that have positive discharge outcomes, and we obtain perfectly balanced datasets. This is performed for both *Drugs* and *Features* for each timestep. One may observe that the augmented datasets are significantly reduced, but we do this action willingly to make sure that the model results are attributable to the model itself and do not suffer from poor data. We still do acknowledge that results from synthetic datasets may not be fully representative of the original datasets.

4.2 Networks and Evaluation

To solve the problem of multi-stage classification, we construct an Ensemble pipeline comprising three networks; *Feature* prediction network, *Drug* prediction network, and *Output*

prediction network. Each network is a Feedforward Neural Network (FNN) model. The first network predicts patient features at the next stage, while the second network predicts patient drugs at the next stage. Each network takes a concatenated matrix of features and drugs from the previous stage as an input, then predicts features, drugs, or output depending on the network. Since learning is divided into multiple stages, which represent logical groupings, such as recovery periods in terms of time, using this approach allows obtaining feature and drug predictions solely based on the features and drugs of the previous step, allowing for more targeted learning. The *Feature* network essentially performs linear regression fitting over the stages, while *Drug* and *Output* networks perform logistic regression fitting, predicting probabilities as a result of sigmoid activation. Although we deal with 3 stages, this process may iteratively continue for i stages until the last stage, where the features and drugs at stage $n-1$ are combined and trained in the *Output* network to predict the output. The pipeline can be seen in Fig. 1. The parameters can be observed in Table 3.

Table 3. Pipeline parameters.

Parameters	**Values**
Folds	5
Optimizer	Adamax
Loss Function	MSE Loss
Epochs	50
Batch Size	100
Learning Rate	0.01

We train the full pipeline using 5-fold cross-validation with shuffled observations, however, shuffled observation IDs are still the same across stages to make sure that patient's result is trained against the results from other stages. Such validation means that 80% of the observations are held for training, while the remaining 20% is for validation, and this is performed 5 times. The parameters of the main network are reset for each fold. For all three networks, we use an *MSE* loss function, which provides the best learning results. In a single forward pass, which for each stage s predicts outputs for stage $s+1$, including the final stage, we use an Adamax optimizer that simultaneously optimizes the parameters of the networks for all three stages. We evaluate learning using *recall* and *F1-scores* on drug prediction in Stages 2 and 3 and on output in Stage 3. The results are presented as averages of folds for drugs and output, and each fold average is an average of batches.

Table 4. Multi-stage evaluation results on test folds for a given run.

	Drug Prediction Results		**Output Prediction Results**
Metrics	Stage 2	Stage 3	Output
Recall (fold average)	99.09%	99.98%	99.41%
F1-score (fold average)	98.87%	94.55%	66.37%

5. Results

Model training results can be seen in Fig. 2. We see drastically decreasing loss when predicting drugs and more oscillating loss when predicting the output. *T1-T2* means we predict timestep 2 drugs using timestep 1 input data, and *T2-T3* means we predict timestep 3 drugs using timestep 2 input data. The test evaluation results can be seen in Table 4. The model can predict drugs across stages with significantly high accuracy. There is room for improvement in output network prediction. The results validate the model training results shown in Fig. 2, where the learning of the output network is not very smooth. It should be noted that evaluation results may change depending on the run as a result of cross-validation; however, the results should be close across the runs.

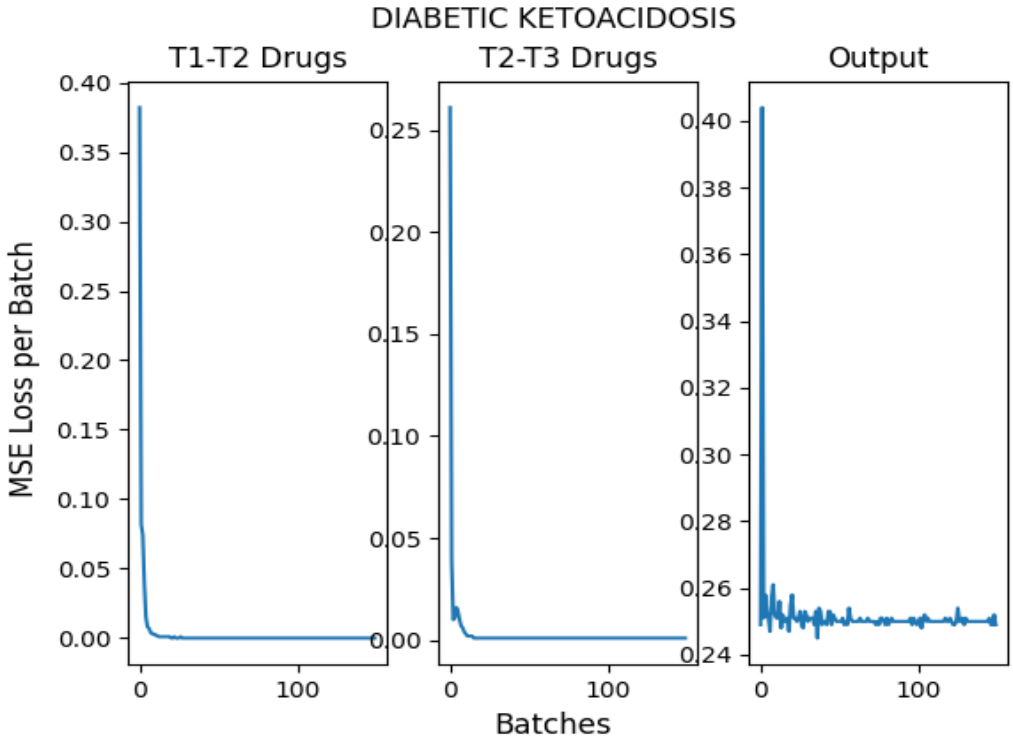

Fig. 2. Training results.

6. Limitations

We identify one main limitation of the paper. Stage-based and generated synthetic datasets do not allow us to benchmark our approach and results with existing similar studies. Although we provide evaluation results for these datasets, verifying the validity of the approach with existing studies could add significant value to the work. Providing benchmark datasets with the logic described in the paper can be part of future work.

7. Conclusion

In this work, we proposed an FNN-based pipeline that combines ensemble learning and iterative classification for modeling a multi-stage drug prescription procedure. The goal of

the approach was to make treatment assignments a dynamic rather than a one-stage static process by utilizing multiple predictor networks. Preprocessed and synthetic datasets for each stage are also provided. In addition, we also evaluated the performance of the whole approach based on how well the pipeline predicted drugs and the output on the testing folds. Although the results are promising, we acknowledge that the evaluations are based on synthetically derived datasets, which may affect the findings.

As part of future work, we can try to achieve class balance for bigger datasets, provide dataset benchmarks, improve the output prediction network, and come up with other logical groupings of a stage besides time. We also acknowledge that treatment predictions may become credible only after proper medical testing and validation.

References

[1] K. Gishyan, "Drug-treatment generation combinatorial algorithm based on machine learning and statistical methodologies", *Open Journal of Applied Sciences*, vol. 13, no. 4, pp.548–561, 2023.

[2] K. Gishyan, H. Sahakyan and L. Ananyan, "Time-stage driven pathfinding framework for optimized medical treatments", *Cogent Engineering*, vol. 10, no. 1, pp.2249258, 2023.

[3] Y. Ren and L. Zhang and P. H. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions", *IEEE Computational Intelligence Magazine*, vol. 11,no. 1,pp. 41–53, 2016.

[4] T. Dietterich, "Ensemble methods in machine learning", *International Workshop on Multiple Classifier Systems*, pp. 1–15, 2000.

[5] E. Ke Tang, P.N. Suganthan and X.Yao, "An analysis of diversity measures", *Machine Learning*, vol. 65,no. 1, pp. 247–271, 2006.

[6] Y. Ren, P.N. Suganthan, N. Srikanth, "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods, *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 236-244, 2014.

[7] A. Chandra and X. Yao, "Multi-objective ensemble construction, learning and evolution, *In Proc. PPSN Workshop Multi-objective Problem Solving from Nature (Part 9th Int. Conf. Parallel Problem Solving from Nature: PPSN-IX)*, pp. 913, 2006.

[8] M. Asafuddoula, B. Verma and Zh. Mengjie, "A divide-and-conquer-based ensemble classifier learning by means of many-objective optimization", *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 762-777, 2018.

[9] X. Zeng, W. Ouyang and X. Wang, "Multi-stage Contextual Deep Learning for Pedestrian Detection", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 121–128, 2013.

[10] E. Alpaydin and C. Kaynak "Cascading classifiers", *Kybernetika*, vol. 34, no. 4, pp. 369–374, 1998.

[11] Zh. Sun, Y. Wang, T. Tan, and J. Cui, "Improving iris recognition accuracy via cascaded classifiers", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 3, pp. 435–441, 2005.

[12] Y. Wang and J. Patrick, "Cascading classifiers for named entity recognition in clinical notes", *Proceedings of the Workshop on Biomedical Information Extraction*, pp. 42–49, 2009.

[13] M. Sabokrou, M. Fayyaz, M. Fathy and R. Klett, "Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes", vol. 26, no. 4, pp. 1992–2004, 2017.

[14] Md. S. H. Shovon, Md. J. Islam, M. N. Ali Khan Nabil, Md Mohimen Molla, Akinul Islam Jony and MF Mridha, "Strategies for enhancing the multi-stage classification performances of her2 breast cancer from hematoxylin and eosin images", *Diagnostics*, vol. 12, no. 11, pp. 2825, 2022.

[15] I. I. Esener, S. Ergin, T. Yuksel, et al., "A new feature ensemble with a multistage classification scheme for breast cancer diagnosis", *Journal of Healthcare Engineering*, vol. 2017, doi:10.1155/2017/3895164, 2017.

[16] Y. Isler, A. Narin, M. Ozer, and M. Perc, "Multi-stage classification of congestive heart failure based on short-term heart rate variability", *Chaos, Solitons & Fractals*, vol. 118, pp. 145–151, 2019.

[17] Y. Kutlu and D. Kuntalp"A multi-stage automatic arrhythmia recognition and classification system", *Computers in Biology and Medicine*, vol. 41, no. 1, pp. 37–45, 2011.

K.R. Kruthika (Research Scholar), Rajeswari (Professor), H.D. Maheshappa (Professor),

[18] K.R. Kruthika, Rajeswari, H. D. Maheshappa, "Multistage classifier-based approach for Alzheimer's disease prediction and retrieval", *Informatics in Medicine Unlocked*, vol. 14, pp. 34–42, 2019.

[19] S. Manimurugan, S. Almutairi, M. M. Aborokbah, C. Narmatha, S. Ganesan, N. Chilamkurti, R. A. Alzaheb and H. Almoamari, "Two-stage classification model for the prediction of heart disease using IoMT and artificial intelligence', *Sensors*, vol. 22, no. 2, pp. 476, 2022.

[20] P. Tang, X. Yan, Y. Nan, Sh. Xiang, S. Krammer and T. Lasser, "FusionM4Net: A multi-stage multi-modal learning algorithm for multi-label skin lesion classification", *Medical Image Analysis*, vol. 76, pp. 102307, 2022.

[21] GB Praveen and A. Agrawal, "Multi stage classification and segmentation of brain tumor", *Proceedings International Conference on Computing for Sustainable Global Development (INDIACom)*,pp. 1628–1632, 2016.

[22] N. Sambyal, P. Saini, R. Syal and V. Gupta,"Aggregated residual transformation network for multistage classification in diabetic retinopathy", *International Journal of Imaging Systems and Technology*,vol. 31, no. 2, pp. 741–752, 2021.

[23] J. Oh, Ch. Park, H. Lee, B. Rim, Y. Kim, M. Hong, J. Lyu, S. Han and S. Choi, "OView-AI supporter for classifying pneumonia, pneumothorax, tuberculosis, lung cancer chest X-ray images using multi-stage superpixels classification", *Diagnostics*, vol. 13, no. 9, pp. 1519, 2023.

[24] N. Patki and R. Wedge and K. Veeramachaneni, "The synthetic data avult", *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399-410,

[25] L.Xu, M. Skoularidou, A. Cuesta-Infante and K. Veeramachaneni, "Modeling tabular data using conditional gan", *Advances in Neural Information Processin Systems*, vol. 32, 2019.

Փուլային դասակարգման սխեմա՝ օպտիմալ բուժում ստանալու նպատակով

Կարեն Մ. Գիշյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
e-mail: karengishyan.res@outlook.com

Ամփոփում

Գոյություն ունեցող մեքենայական ուսուցման և խորը ուսուցման պարամետրերում դասակարգման և ռեգրեսիայի կանխատեսման խնդիրները կարող են ներկայացվել որպես գործընթաց, որտեղ մոդելի ելքը հիմնված է մեկ փուլային մուտքագրման վրա: Իրական կյանքի սցենարների մեծ մասում հիվանդի համար ցանկալի բժշկական վիճակին հասնելը կարող է ներառել դեղերի նշանակման խնդիրների դինամիկ լուծում՝ հիմնված մուտքային տվյալների վրա տարբեր փուլերում, որտեղ յուրաքանչյուր փուլ տրամաբանական խմբավորում է, ինչպիսիք են՝ ժամանակի բաժանումը, ինտենսիվ թերապիայի բաժանմունքում մնալը և այլն: Տվյալ փուլի տվյալները ներկայացնում են վերականգնման առաջընթաց և կարող են էապես տարբերվել նախորդ և ապագա փուլերի տվյալներից: Թեև մեկ մոդելը կարող է լուծել առաջադրանքը, ուսուցման բազմափուլ ընթացակարգը կարող է ավելի հարմար լինել: Այս խնդիրը լուծելու համար մենք առաջարկում ենք FNN-ի վրա հիմնված մոդելների միավորման մոտեցում՝ կանխատեսելու այն դեղամիջոցները, որոնք հիվանդը պետք է ստանա վերականգնման գործընթացի յուրաքանչյուր փուլում: Վերջնական բժշկական դուրսգրման վայրը կանխատեսվում է դեղերի և վիճակի տվյալների հաջորդական կանխատեսումների արդյունքում: Այս աշխատանքում մենք համատեղում ենք մոդելային միավորումը և բազմափուլ իտերատիվ ուսուցումը՝ օպտիմալ դեղերի նշանակման առաջադրանքը լուծելու համար՝ որպես ներդրում առկա գրականության մեջ:

Բանալի բառեր՝ Փուլային դասակարգում, բուժման օպտիմալացում, մոդելների միավորում, մեքենայական ուսուցում:

Многоступенчатая схема классификации для оптимизации медицинского лечения

Карен М. Гишян

Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
e-mail: karengishyan.res@outlook.com

Аннотация

В большинстве реальных сценариев достижение желаемого медицинского состояния пациента может включать динамическое решение задач назначения лекарств на основе входных данных на разных этапах, где каждый этап представляет собой логическую группировку, такую как разделение временных шагов, пребывание в отделении интенсивной терапии и т. д. Данные на данном этапе представляют собой прогресс восстановления и могут принципиально

отличаться от наборов данных с предыдущих и будущих этапов. Хотя одну модель может решить задачу, многоэтапная процедура обучения может оказаться более подходящей. Для решения этой задачи мы предлагаем основанный на FNN ансамблевый подход для прогнозирования лекарств, которые пациент должен получать на каждом этапе процесса выздоровления. Окончательное место медицинской выписки прогнозируется в результате последовательного прогнозирования препаратов и особенностей. В этой работе мы объединяем ансамбль моделей и многоэтапное итеративное обучение для решения задачи создания оптимального рецепта на лекарства в качестве вклада в существующую литературу.

Ключевые слова: Многоэтапная классификация, оптимизация лечения, ансамбль моделей, машинное обучение.