Workflow model for big data analysis and visualization

Stella Vetova
Department Informatics
Technical University of Sofia
Sofia, Bulgaria
vetova.bas@gmail.com

Abstract—The presented report includes workflow model for biomedical data. The proposed model is a solution for the case study of two-dimensional data classification. The experiments are conducted using distance computing techniques: Euclidean distance, Manhattan Distance, Cosine Distance. The result values show the percentage of accurately classified images for the three distance measures. In addition, the paper also presents a workflow model as a solution for prediction problems. The experiments are performed to compute efficiency and probability of heart disease occurrence using Logistic Regression, Decision Tree, Random Forest and Naïve Bayes techniques. The results show the advantages and disadvantages of the included methods.

Keywords—biomedicine, bioinformatics, big data workflow, 3D model, workflow analysis, biomedical data analysis and visualization

I. INTRODUCTION

The development of high technology, science, business, medicine, biomedicine [1], bioinformatics [2], agriculture, biology [3], etc. leads to the emergence of big data. It is a generated heterogeneous amount of data used for future processing. Big data has four basic features: complexity, variety, volume and application opportunities. In the field of medicine big data is gained on the base of the separate components of clinical workflow such as laboratory results, clinical test and patients' exams, symptoms data captured by the means of telemedicine, etc. The technical apparatuses needed for image information gathering varies in a great range. Up to date ones of the most often used methods are: X-Ray, Ultrasound, Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) producing images in DICOM format. In this case, big data requires technical support and devices capable of fast data processing and interpretation which is hard to be reached using the traditional tools. Thus, being in help of the physicians, workflows for big data analysis and visualization can be designed integrating the methods of Artificial Intelligence (AI) and Machine Learning (ML).

Workflow for science is a sequence of functions defined to perform single task. United in a system scientific workflow computes complex tasks and represents a complex software application which may require a long period of time to do the calculations [4]. Its basic stages, specifically in the biomedicine, bioinformatics, biology domain, etc. include data visualization and analysis based on the principles of segmentation, diagnosis and therapy. Using the obtained data, workflows are applied for complex automatic analysis in the strive to improve the interpretation and reporting, reducing time and providing ease of decision making in the process of disease diagnosis and treatment, classification and detection.

On the other hand, as a part of hosting (Amazon), big data is also protected against threats in the cloud [5] and the network [6], [7] through encryption algorithms.

The following paper presents a brief overview of the methods and algorithms for big data workflow analysis, visualization and interpretation in the field of biomedicine. It is organized as follows. Section 2 is divided into two subsections: Subsection A introduces methods and algorithms for big data workflow analysis in biomedicine and subsection B introduces methods and algorithms for visualization and interpretataion of big data workflows in biomedicine. Section 3 presents the workflow models for biomedical data processing for the cases of Distance-Based Image Classification and Prediction-Based Workflow Model along with the experiments performed and obtained results in subsections A and B respectively. Section 4 concludes the paper.

II. RELATED WORK

A. Methods and Algorithms for Big Data Workflow Analysis in Biomedicine

The applied methods for ML models design are based on groups of methods referring clustering, instance computation, decision making, deep learning (DL) [8], [9] which is largely used in finding solution of the biomedicine problems, Bayesian models, image compression, etc. More particularly the groups include k-Means partitioning algorithm, k-medoids algorithm [10], k-Nearest Neighbor (KNN), hierarchical clustering [11] including Unweighted Pair Group Method with Arithmetic Mean (UPGMA), the NeighborJoining (NJ) method, and the Fitch and Kitsch method [10], Self-Organizing Map (SOM), Conditional Decision Trees, KD-Trees data structure [12], Deep Boltzmann Machine (DBM), Convolutional Neural Network (CNN) [13], Support Vector Machine (SVM) [14]. In regard to image compression losseless image compression is preferable since in medicine domain the slightest detail is of great importance. This compression group includes *entropy encoding*, such as the Shannon-Fano algorithm, Huffman coding, arithmetic coding, Lempel-Ziv-Welch algorithm [14].

In addition, Eric P. Xing, Qirong Ho, Pengtao Xie, Dai Wei [15], make an overview of the design principles of the distributed ML systems. They point out iterative-convergent ML group of algorithms including the structured sparse

regression ML algorithm family and particularly Lasso regression. The second algorithm they discuss is Latent Dirichlet allocation from the graphical models ML algorithm family. In their work the authors also pay attention to the properties of ML programs such as error tolerance necessary for ML programs to avoid errors, dependency structure, non-uniform convergence as properties providing balance between speed, programmability and correctness. Furthermore, the authors discuss the principles of ML system design such as: dependency structures in ML programs, scheduling in ML programs, compute prioritization in ML programs, balancing workloads in ML programs, structure aware parallelization. As a part of the continuous communication strategy update prioritization, parameter storage and communication topologies are described.

In its traditional form a workflow design starts with the data inputs and continues through a sequence of functions which computations result in data outputs. A main drawback of the workflows is the lack of opportunity for reusing workflows. Typical workflow design tools include: Galaxy [16], [17], [18], the sharing workflow platform myExperiment, Taverna [19], [20] MapReduce, Hadoop, Spark, GraphLab, Pregel [15], Closha [21], Preglix [22], GraphX [23], and PEGASUS [24], [25], Kepler, Chipster [26], Disco [27], Tavaxy [28], e UGENE Workflow Designer [29].

In [25] Riazi and Norris present a toolkit for workflow design called GaphFlow. It is based on tools compatible with Galaxy and allows conducting science experiments with complex data where Spark components are used for the workflow design. To process data, the proposed architecture relies on a cluster system using a Spark master node. The GraphFlow includes I/O tools, tools for performing graph analysis, relational tools, plotting tools. A feature of GraphFlow is returning a log output file. It is a text file compatible with Galaxy containing output dataframes and the tool execution log. The architecture is designed to convert single file data into dataframes and vice versa. It has the ability to use the metafile constructed on the base of the users' data for the cases when users upload their data to the cloud storage. In this case the MetaLoader component of I/O tools is provided. Furthermore, to generate and process graphs the proposed architecture uses nine specified algorithms such as: GraphGen, PageRank used for rank assessment, DegreeCount for degree computation, TriangleCount used for triangle count computation, SubGraph applied for the cases when a subgraph of the original graph is needed to be constructed, LargestCC used to output the designed subgraph of the largest connected elements of the original one. For the cases of clustering the GraphFlow is based on the algorithms GraphCluster, ClusterEval for assessment of the clustering quality, GraphCoarsen for simplifying a big graph, PIC, spectral clustering, label propagation. On the other hand, components of the relational tools are responsible for the transformation of dataframes. Relational tools are based on SQLwhen it comes to query. The statistics tools of the GraphFlow has the task to gather statistics data from the dataframe where cumulative density function (CDF) is used for data distribution analysis. As plotting tools ScatterPlot and HistogramPlot are added[25].

Spjuth [26] states that for the bioinformatics data processing the scripting languages Bash, Perl [30] and Python are most often used for automate analysis. In addition, a private cloud is constructed to contain images and the option for work with workflow platforms Galaxy, Chipster and GPCR-ModSim. For the feature counting and quality assessment tasks in RNA-seq analysis a solution is offered based on the extension of HTSeq packet using the tools of Hadoop and MapReduce. On the other hand, for running pipelines the author points out that Make is applied for both cases of use local and cluster. Snakemake is chosen to be applied for scientific tests because of its ease of workflow transfer to a cluster, the option for parallelization, Bash code portability, integration with such programming languages as Python, etc. Furthermore, Chipster is declared to be used for the scenario of analysis of RNA-seq data and ChIP-seq data.

In [31] performance evaluation of scientific workflows is discussed. A lightweight metric for evaluation of synthetic workflow performance is proposed and tested in two different cases of in situ workload execution using GROMACS molecular dynamic application.

Michael T.Krieger et al. [32] discuss two ideas. The first one refers to the notion to construct full cloud stack including IaaS, PaaS, SaaS on an open source technology. In addition, the authors propose a strategy to design workflows using Galaxy framework. The feasibility and performance guaranteed using the proposed method are demonstrated through applications for the cases of bioinformatics and biomedicine.

TABLE I. METHOS, ALGORITHMS AND TOOLS FOR BIG DATA ANALYSIS

Methods	**Clustering**	k-medoids algorithm;
		k-Means partitioning algorithm;
		k-Nearest Neighbor (KNN);
		Unweighted Pair Group Method with Arithmetic Mean (UPGMA);
		Neighbor-Joining (NY) method;
		Fitch and Kitsch method;
	Instance computation	Euclidean distance;
		Hamming distance;
		Manhattan distance;
		Minkowski distance;
	Decision Making	Self-Organizing Map (SOM);
		Conditional Decision Trees;

	Deep Learning	Convolutional Neuron Networks (CNN);
		Deep Boltzmann Machine;
	Bayesian models	Regularized Bayesian models;
		Nonparametric Bayesian models;
	Relational DBs	SQL;
	Data Distribution Analysis	Cumulative Distribution Analysis (CDF);
ML Groups of algorithms	**The structured sparse regression ML algorithm family**	Lasso regression;
	The graphical models ML algorithm family	Latent Dirichlet allocation;
Workflow platforms and design tools	Galaxy;	
	myExperiment	
	Taverna	
	MapReduce	
	Hadoop	
	Spark	
	GraphLab	
	Pregel	
	Closha	
	Preglix	
	Pegasus	
	Kepler	
	Chipster	
	Mike	
	GraphFlow	
	Gromacs	
Storage Environment	Cloud storage;	
	Local storage	
Programming Languages	Bash	
	R	
	Perl	
	Python	

B. Methods and Algorithms for Visualization and Interpretataion of Big Data Workflows in Biomedicine

The emergence and growth of big data, especially in the fields of medicine, biomedicine, bioinformatics, require its visualization. This provokes the design of visualization tools. In workflow analysis there are four basic stages of data processing. They include: data storage, data processing, querying, data analysis and classification, visualization [33].

Julien Wist [34] presents a web solution for big data processing and visualization working in offline mode. To this end, data and a constructed function for visualization are needed where the programming languages for data analysis and for visualization can be different. The visualizer package includes also the general concept to process and change the input data and to store the result as an output variable. In addition, for the cases when relation between the objects exists it is possible to add the same unique tag and thus tracing the position of the mouse. The proposed approach is based on Java Script.

Rubens et al. [35] present an open-source web tool for bioimage analysis workflows called BIFLOWS. In addition, they illustrate a comparison of seven nuclei segmentation workflows. BILFLOWS is dedicated to work with aforetime annotated multidimensional microscope visual data. It is a framework with the following main features: first, import of image databases containing annotation and their organization as bioimage analysis tasks. Second, bioimage analysis workflow encapsulation, third, image processing and visualization in combination with the results and finally, automatic evaluation of the workflows performance.

In [36] the authors present FAN-C which is a framework combining matrix generation, analysis and visualization in the field of bioinformatics. To perform fast matrix access and Hi-C matrix transformations the framework is designed with hierarchical storage architecture. Also, FAN-C enables the import of variety of text-based matrix inputs. In regard to pipelining, the framework supports the option for adaptation of the automated FASTQ-to-matrix pipeline to the requirements of the scientific experiments and Hi-C analysis performed. Furthermore, the framework enables the running of the pipeline functions separately and enables individual setting for each of them. In addition, the users can choose filters through the Python API which is a component of the framework. It also offers automatically generated diagnostic plots with filtering statistics which task is informing the user of issues.

Bergenstråhle et al. [37] present the R-based STU utility for bioinformatics. It has the ability to identify spatial patterns alignment of tissue images and visualization. As input data the utility works with RNA count and images. The functionality of STU utility covers data analysis, image processing stage and visualization. During the process of image processing a masking procedure to eliminate the background is used. The method applied transforms the low-level image representation into superpixels. Then, the K-means clustering algorithm is performed to classify the areas inside and outside of the tissue. In addition, an iterative closest point algorithm (ICP) is used for the automatic alignment function. For the cases when the described function fails manual image alignment is provided. In addition, the aligned images can be organized to construct a 3D model of the tissue on the base of cell segmentation thus, capturing the its morphological structure. As a method for

identification and extraction of neighboring capture-spots and as a test to rank genes Non-negative Matrix Factorization is used for analysis performed.

Wollman et al. [16] propose workflows for image processing and analysis for large scale experiments in the field of biology. The authors use KNIME and Galaxy as workflow systems for the proposed method.

In Fig. 1 and Fig. 2 the visualization tools and methods for biomedical image processing are graphically illustrated.

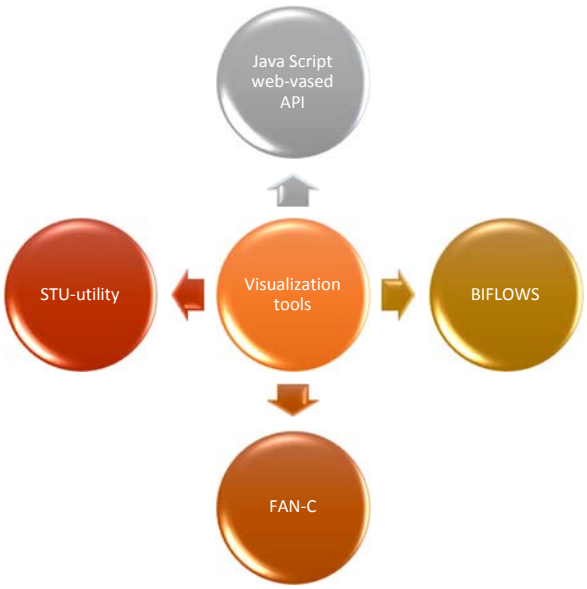

Fig. 1. Big data visualization tools

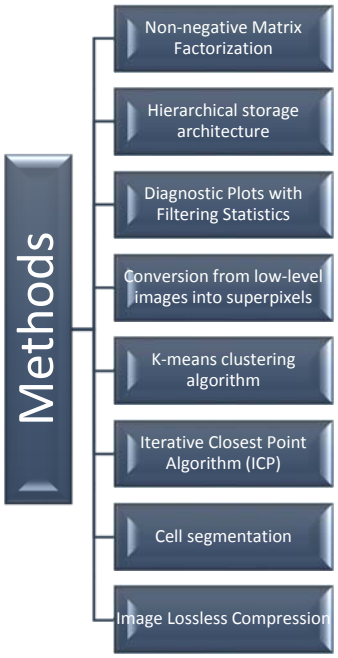

Fig. 2. Methods for biomedical image processing

III. Workflow Models for Biomedical Data Processing

For the goal of the experiments conducted, two test image databases are used: the first one contains 758 Covid images in grayscale color space. The images are classified into two groups: Covid positive and Covid negative with 361 and 397 images respectively. They are distinguished for JPEG and PNG format. The second one is table-based structured in 14 attributes describing the current state of patients with heart disease.

The tests are performed in accordance with the tasks for a framework design for biomedicine via the software for data visualization, machine learning, data mining and data analysis Orange v. 3.27 on personal computer with the following configuration: Intel (R) Core (TM) 2 Duo 2,40 GHz, 64-bit Operating System. The experiments present two models of workflows for the goal of:

1. image classification based on distance computing with three of the most often used metrics: Euclidean distance, Manhattan, Cosine distance and efficiency evaluation using cross validation method presented in Fig.3.;
2. decision tree-based data prediction and logistic regression-based prediction techniques presented in Fig. 13.

With regard to this we computed the percentage of correctly classified instances, ROC AUC area, Classification Accuracy (CA), F1, Precision, Recall and Confusion matrix for the first workflow model and ROC AUC area, Classification Accuracy (CA), F1, Precision, Recall and the occurrence probability for the second workflow model.

A. Distance-Based Workflow Model

For the workflow performance, first the Covid test image database is loaded (Fig. 4). It can be visualized using a visualization tool and its metadata can be displayed using the module Data Table. In the image preprocessing stage features extraction process for image description through deep network embedding with the Image Embedding component is conducted. The obtained results can be displayed with Data Table where along with image metadata, the image feature vectors are listed (Fig. 5). For the phase of image classification, first cross validation on the base of logistic regression method is designed. As a result, the efficiency is computed by the measures: ROC AUC area, Classification Accuracy (CA), F1, Precision, Recall and Confusion matric for results interpretation. Similarity computing is performed on the base of three components: Distances which allow the choice of distance measure, Hierarchical Clustering, Image viewer for classification result visualization.

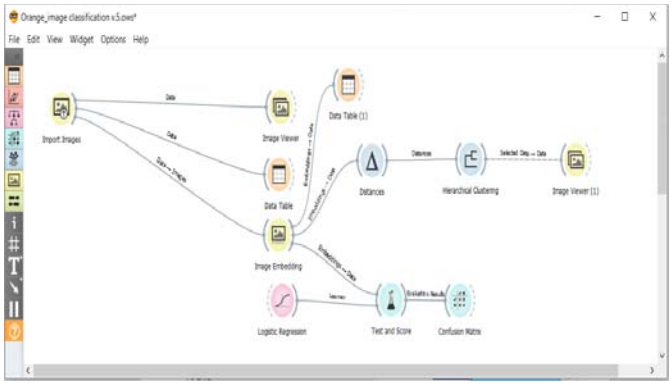

Fig. 3. Distance-based image classification workflow with Orange

Fig. 4. Covid test image database

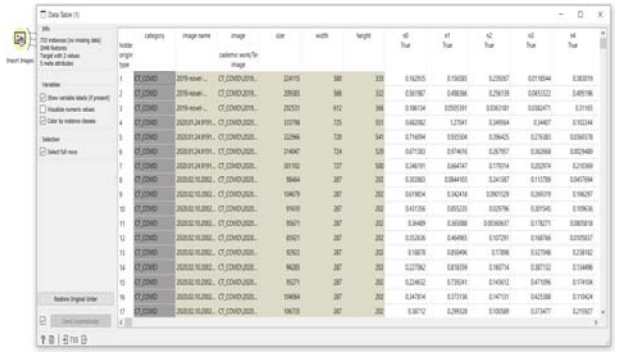

Fig. 5. Image meta data and feature vectors

According to the results obtained on the base of logistic regression the efficiency evaluation demonstrates high values with AUC = 0,852; CA=0,779; F1=0,779, Precision=0,779; Recall=0,779 as presented in Fig. 6. Fig. 7 presents the number of correctly predicted number of Covid positive and Covid negative cases. In addition, Fig. 8 presents the result of image classification based on Euclidean distance. The result shows nearly 75% correctly classified items graphically displayed in Fig. 9. Similarly, in Fig. 10, Fig. 11 and Fig. 12 the results of Manhattan and Cosine distance are presented. The comparative analysis based on the three distance metrics show that the classification results are highest using Cosine distance where the percent of correctly classified items is approximately 93.75% vs 87.5% for Manhattan.

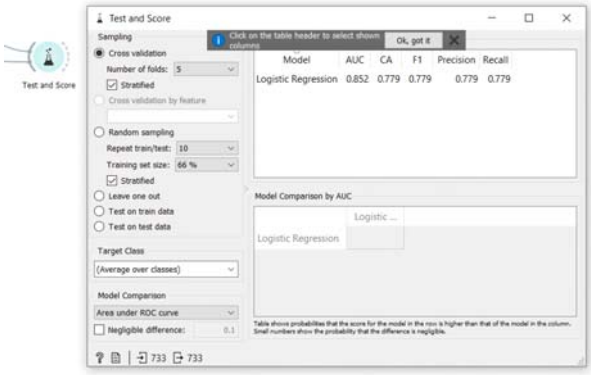

Fig. 6. Efficiency evaluation results for distance-based workflow model

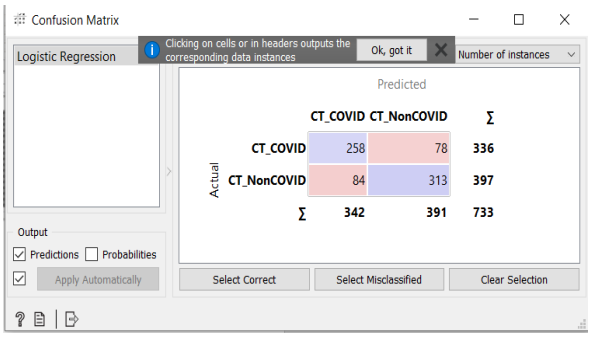

Fig. 7. Confusion matrix

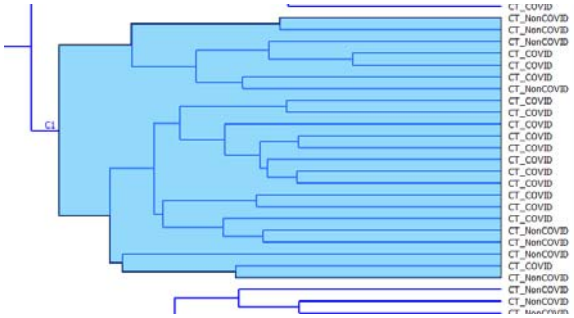

Fig. 8. Covid image classification using Euclidean distance

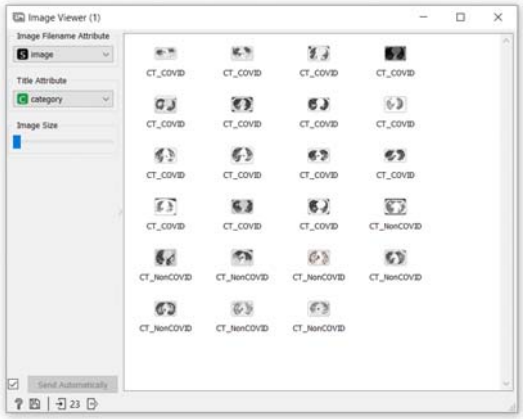

Fig. 9. Visualization of Euclidean-based Covid image classification

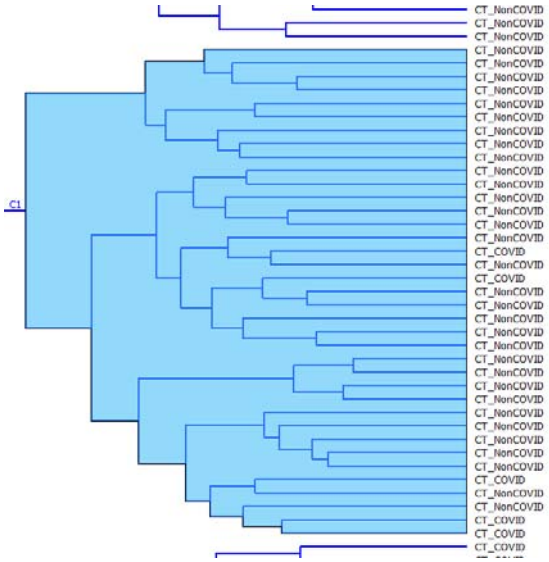

Fig. 10. Covid image classification using Manhattan distance

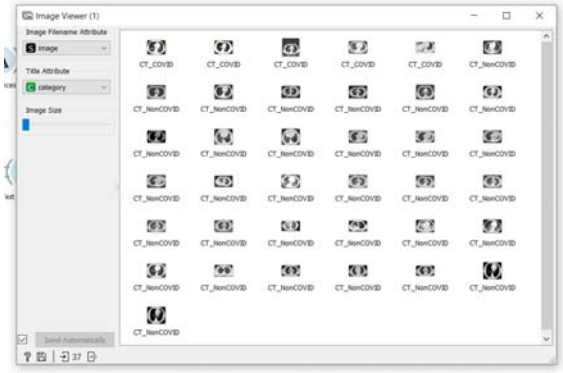

Fig. 11. Visualization of Manhattan-based Covid image classification

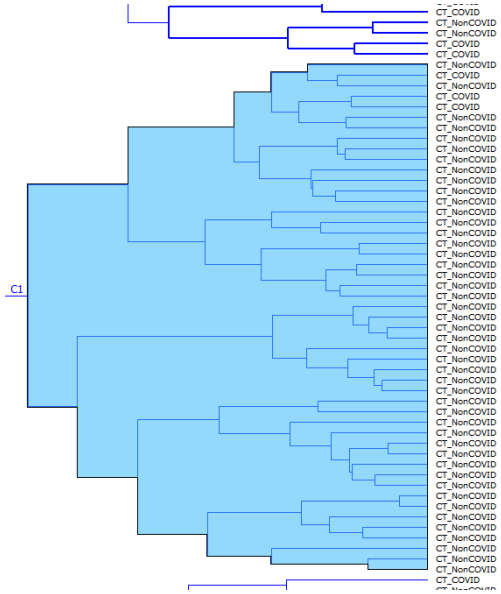

Fig. 12. Covid image classification using Cosine distance

B. *Prediction-Based Workflow Model*

The goal of the second developed workflow (Fig.13) is to predict the probability of heart disease for a patient. First, the test data base is loaded and data is displayed using the Data Table component. To perform classification and realize the stage of decision making, Decision Tree is added and the result is visualized as Fig. 14 shows. To make a prediction correctly the prediction model is built using four methods. The first one is built on Decision Tree connected to the Prediction component. The second one is based on Logistic Regression connected to the Prediction component performing the process of prediction. Similarly, the Random Forest and Naïve Bayes are connected to the prediction component to compute the probability if heart disease. The final prediction results for the four approaches are graphically illustrated in Fig. 15. The results show similar probability values for the Decision Tree, Logistic Regression and the Random Forest. Naïve Bayes demonstrates higher probability values approaching 1. In addition, the results obtained on the base of the described methods the efficiency evaluation is computed. According to the results, Logistic Regression demonstrates the highest values with AUC = 0,908; CA=0,845; F1=0,844, Precision=0,846; Recall=0,845 followed by Naïve Bayes method as follows: AUC = 0,907; CA=0,835; F1=0,835, Precision=0,835; Recall=0,835. The Decision Tree demonstrates the lowest efficiency values as shown in Fig. 16.

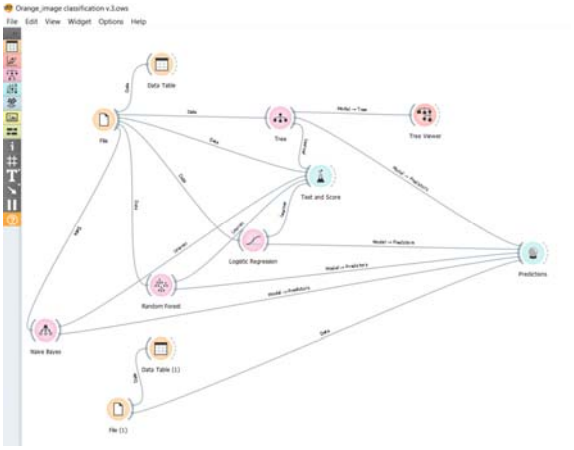

Fig. 13. Prediction-Based Workflow Model

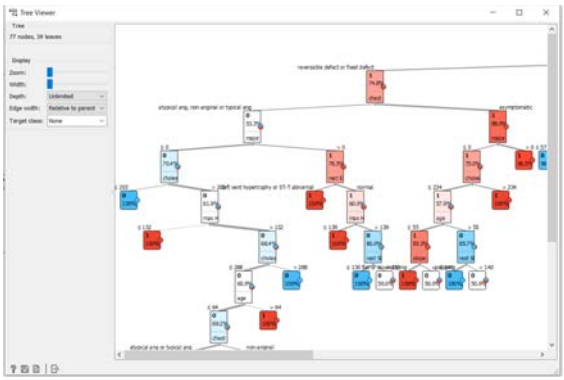

Fig. 14. Data classification using decision tree

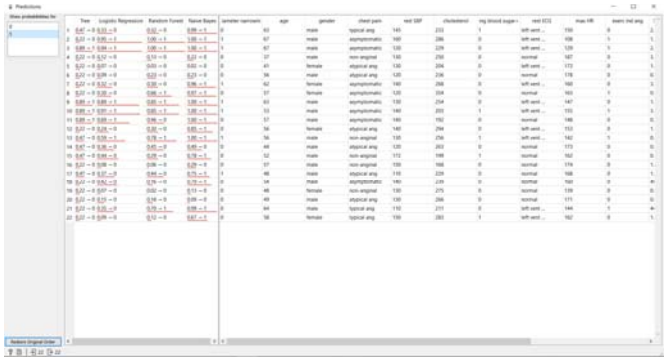

Fig. 15. Probability computation using decision tree, logistic regression, random forest and Naïve Bayes

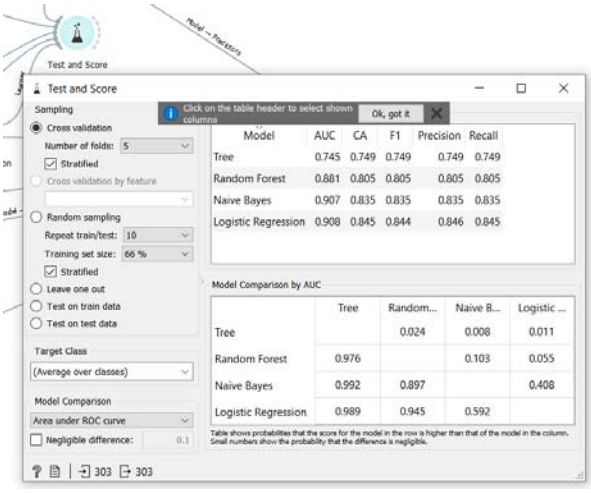

Fig. 16. Efficiency evaluation results for prediction-based workflow model

CONCLUSION

In the presented paper two workflow models for biomedical data processing are described. The first one offers solution to the problems for the case study of two-dimensional data classification. The performed experiments are based on three of the most used distance computing techniques: Euclidean distance, Manhattan Distance, Cosine Distance. The experiments demonstrate best results for image classification using Cosine distance with nearly 93.75% accurately classified images followed by the result for Manhattan distance – approximately 87.5% and 75% for Euclidean Distance. In addition, the applied Logic Regression function provides the option for modelling the probability of occurrence of the two image classes where the correct predicted instances are marked in blue in the confusion matrix.

The second workflow is designed to offer solution to the prediction problems. In this case, the probability of heart disease occurrence is computed through Logistic Regression, Decision Tree, Random Forest and Naïve Bayes techniques. Regarding the disease probability computation the Decision Tree, Logistic Regression and the Random Forest generate similar results unlike Naïve Bayes which demonstrates higher probability values approaching 1. On the other hand, the Logistic Regression shows the highest values for efficiency followed by Naïve Bayes, the Random Forest and the Decision Tree.

ACKNOWLEDGMENT

Primary funding for the presented work was provided by the National Science Fund, Ministry of Education and Science, Republic of Bulgaria under contract KP-06-N37/24, research project "Innovative Platform for Intelligent Management and Analysis of Big Data Streams Supporting Biomedical Scientific Research".

REFERENCES

[1] Kordasht, H., Hasanzadeh, M., "Biomedical analysis of exosomes using biosensing methods: recent progress," Journal Analytical Methods, Issue 22, 2020, pp. 2795-2811.

[2] Carretero, J., Krefting, D., "New Parallel and Distributed Tools and Algorithms for Life Sciences," Future Generation Computer Systems, Vol. 112, November 2020, pp. 1174-1176.

[3] V. Gancheva, "SOA based multi-agent approach for biological data searching and integration," International Journal of Biology and Biomedical Engineering, ISSN: 1998-4510, Vol. 13, 2019, pp. 32-37.

[4] Talia, D.. "Workflow Systems for Science: Concepts and Tools." International Scholarly Research Notices 2013 (2013): 1-15.

[5] Ivanov, I., "Basic Cloud Security Threats," Proceedings of Annual University Science Conerence, vol. 6, Veliko Tarnovo, Bulgaria, May 2020, pp. 143 – 147.

[6] Ivanov, I., "Entry Points for Cyberatacs," International Science Conference "Wide Security", vol. 2, New Bulgarian University, Sofia, March, 2020m pp. 336 – 341, ISBN 978-619-7383-19-5.

[7] Ivanov, I., "Analysis of vulnerabilities in web applications," Proceeding of Science Conference "Current Security Issues," Veliko Tarnovo, vol. 6, 2020, pp. стр. 233 – 236. ISSN 2367-7465.

[8] Rodriguez, J., Rinschen, M., Floege, J., Kramann, R., "Big science and big data in nephrology," Kidney International, Vol. 95, Issue 6, 2019, pp. 1326-1337, ISSN 0085-2538.

[9] Jahan, F., Khan, P., Sapkal, R., Vinod V.C, Mehetre, V., "Biomedical Image Analysis and Deep Learning," IOSR Journal of Dental and Medical Sciences (IOSR-JDMS) e-ISSN: 2279-0853, p-ISSN: 2279-0861. Vol. 19, Issue 2 Ser.15 (February. 2020), pp. 32-36.

[10] Lord, E., ,Diallo, A., Makarenkov, V., "Classification of bioinformatics workflows using weighted versions of partitioning and hierarchical clustering algorithms," . BMC Bioinformatics (2015) pp. 16:68.

[11] Chojnowski, G., Walen´, T., Bujnick, J., "RNA Bricks—a database of RNA 3D motifs and their interactions," Nucleic Acids Research, 2014, Vol. 42, Database issue, pp. D123–D131.

[12] Walen´, T., Chojnowski, G., Gierski, P., Bujnicki, J., "ClaRNA: a classifier of contacts in RNA 3D structures based on a comparative analysis of various classification schemes," Nucleic Acids Research, 2014, Vol. 42, No. 19 pp. e151.

[13] Cirillo, D., Valencia, A., "Big data analytics for personalized medicine," Current Opinion in Biotechnology, Volume 58, August 2019, Pages 161-167.

[14] Kouanoua, A., Tchiotsopb, D., Kengnea, R., Zephirind, D., Armelea, N., Tchinda, R., "An optimal big data workflow for biomedical image analysis," Informatics in Medicine Unlocked 11 (2018) pp.68-74.

[15] Xing, E., Ho, Q., Xie, P., Wei, D., "Strategies and Principles of Distributed Machine Learning on Big Data," Engineering 2 (2016) pp. 179–195.

[16] Wollmann, T., Erfle, H., Eils, R., Rohr, K., Gunkel, M., "Workflows for microscopy image analysis and cellular phenotyping," Journal of Biotechnology, Vol. 261, 10 November 2017, pp. 70-75.

[17] Garza, L., Veit, J., Szolek, A., Röttig, M., Aiche, S., Gesing, S., Reinert, K., Kohlbacher, O., "From the desktop to the grid: scalable bioinformatics via workflow conversion," BMC Bioinformatics (2016) 17:127.

[18] Kanwal, S., Khan, F., Lonie, A., Sinnott, R., "Investigating reproducibility and tracking provenance – A genomic workflow case study," BMC Bioinformatics (2017) 18:337.

[19] Wolstencroft, K., Haines, R., Fellows, D.,, Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., de la Nieva Hidalga, A., Balcazar Vargas, MP, Sufi, S., Goble, C., "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud," Nucleic Acids Res 2013, 41(Web Server issue):W557–W561.

[20] Hettne, K., Dharuri, H., Zhao, J., Wolstencroft, K., Belhajjame, K., Soiland-Reyes, S., Mina, E., Thompson, M., Cruickshank, D., Verdes-Montenegro, L., Garrido, J., Roure, D., Corcho, O., Klyne, G., Schouwen, R., Hoen, P., Bechhofer, S., Goble, C., Roos, M., "Structuring research methods and data with the research object model: genomics workflows as a case study," Journal of Biomedical Semantics 2014, 5:41.

[21] Ko, G., Kim, P., Yoon, J., Han, G., Park, S., Song, W., Lee, B., "Closha: bioinformatics workflow system for the analysis of massive sequencing data," BMC Bioinformatics 2018, 19(Suppl 1):43.

[22] Bu, Y., Borkar, V., Jia, J., Carey, M., Condie, T., "Pregelix: Big(ger) graph analytics on a dataflow engine," Proceedings of the VLDB Endowment, vol. 8, no. 2, pp. 161–172, 2014.

[23] Gonzalez, J., Xin, R.S., Dave, A., Crankshaw, D., Franklin, M. J., Stoica, I., "GraphX: Graph processing in a distributed dataflow framework," Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI 14, Broomfield, CO, USA, 2014, pp. 599–613.

[24] Kang, U., Tsourakakis, C., Faloutsos, C., "PEGASUS: A peta-scale graph mining system implementation and observations," Proceedings of the 9th IEEE International Conference on Data Mining, ser. ICDM '09, Miami, FL, USA, 2009, pp. 229–238.

[25] Riazi, S., Norris, B., "GraphFlow: Workflow-based Big Graph Processing," 2016 IEEE International Conference on Big Data (Big Data), December 2016, DOI: 10.1109/BigData.2016.7840993, pp. 3336-3343.

[26] Spjuth, O., Bongcam-Rudloff, E., Hernández, G., Forer, L., Giovacchini, M., Guimera, R., Kallio, A., Korpelainen, E., Kanduła, M., Krachunov, M., Kreil, D., Kulev, O., Łabaj, P., Lampa, S., Pireddu, L., Schönherr, S., Siretskiy, A., Vassilev, D., "Experiences with workflows for automating data-intensive bioinformatics," Biology Direct (2015) 10:43.

[27] Merelli, I., Pérez-Sánchez, H., Gesing, S., D'Agostino, D., "Managing, Analysing, and Integrating Big Data in Medical Bioinformatics: Open Problems and Future Perspectives," BioMed Research International Volume 2014, Article ID 134023.

[28] Abouelhoda, M., Issa, Sh., Ghanem, M., "Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support," BMC Bioinformatics 2012, 13:77.

[29] Venco, F., Vaskin, Y., Ceol, A., Muller, H., "SMITH: a LIMS for handling next-generation sequencing workflows," BMC Bioinformatics 2014, 15(Suppl 14):S3.

[30] Raj-Kumar, PK., Liu, J., Hooke, JA., Kovatich, AJ., Kvecher, L., Shriver, CD., Hu, H., "PCA-PAM50 improves consistency between breast cancer intrinsic and clinical subtyping reclassifying a subset of luminal A tumors as luminal B," Sci Rep. 2019 May 28;9(1):7956. doi: 10.1038/s41598-019-44339-4. PMID: 31138829; PMCID: PMC6538748.

[31] AnhDo, T., Pottier, L., Caíno-Lores, S., Ferreira da Silva, R., Cuendet, M., Weinstein, Estrada, T., Taufer, M., Deelman, E., "A lightweight method for evaluating *in situ* workflow efficiency," Journal of Computational Science, Vol. 48, January 2021, 101259.

[32] Krieger, M., Torreno, O., Trelles, O., Kranzlmüller, D., "Building an open source cloud environment with auto-scaling resources for executing bioinformatics and biomedical workflows," Future Generation Computer Systems, Vol. 67, February 2017, pp. 329-340.

[33] Mattingly, W., Kelley, R., Chariker, J., Weimken, T., Ramirez, J., "An iterative workflow for creating biomedical visualizations using Inkscape and D3.js," BMC Bioinformatics 2015, 16(Suppl 15):P10.

[34] Wist, J., "HastaLaVista, a web-based user interface for NMR-based untargeted metabolic profling analysis in biomedical sciences: towards a new publication standard", Journal of Cheminformatics, (2019) 11:75.

[35] Rubens, U., Mormont, R., Paavolainen, L.,Backer, V., Pavie, B., Scholz, L., Michiels, G., Maska, M., U¨nay, D., Ball, G., Hoyoux, R., Vandaele, R., Golani, O., Stanciu, S., Sladoje, N., Paul-Gilloteaux, P., Mare, R., Tosi, S., "BIAFLOWS: A Collaborative Framework to Reproducibly Deploy and Benchmark Bioimage Analysis Workflows," Patterns, Vol.1, Issue 3, 12 June 2020, 100040.

[36] Kruse, K., Hug, C., Vaquerizas, J., "FAN-C: a feature-rich framework for the analysis and visualisation of chromosome conformation capture data," Genome Biology (2020) 21:303.

[37] Bergenstråhle, J., Larsson, L., Lundeberg, J., "Seamless integration of image and molecular analysis for spatial transcriptomics workflows," BMC Genomics (2020) 21:482.