



ACCSE 2020

The Fifth International Conference on Advances in Computation, Communications
and Services

ISBN: 978-1-61208-810-5

September 27th – October 1st, 2020

ACCSE 2020 Editors

Javier Fabra, Universidad de Zaragoza, Spain

ACCSE 2020

Foreword

The Fifth International Conference on Advances in Computation, Communications and Services (ACCSE 2020), held between September 27 – October 1st, 2020, is targeting the progress made in computation, communication and services on various areas in terms of theory, practices, novelty, and impact. Current achievements, potential drawbacks, and possible solutions are aspects intended to bring together academia and industry players

The rapid increase in computation power and the affordable memory/storage led to advances in almost all the technology and services domains. The outcome made it possible advances in other emerging areas, like Internet of Things, Cloud Computing, Data Analytics, Smart Cities, Mobility and Cyber-Systems, to enumerate just a few of them.

We take here the opportunity to warmly thank all the members of the ACCSE 2020 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ACCSE 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ACCSE 2020 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ACCSE 2020 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of computation, communication, and services.

ACCSE 2020 Chairs:

ACCSE 2020 Publicity Chair

Mar Parra, Universitat Politècnica de Valencia, Spain

ACCSE 2020

COMMITTEE

ACCSE 2020 Publicity Chair

Mar Parra, Universitat Politecnica de Valencia, Spain

ACCSE 2020 Technical Program Committee

Safa'a AbuJarour, University of Potsdam, Germany
Kishwar Ahmed, University of South Carolina Beaufort, USA
Maxim Bakaev, Novosibirsk State Technical University, Russia
Abdul Basit, State Bank of Pakistan (Central Bank of Pakistan), Pakistan
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Jagadeesha R Bhat, St. Joseph Engineering College, Mangalore, India
Freimut Bodendorf, Institute of Information Systems - University of Erlangen-Nuremberg, Germany
An Braeken, Vrije Universiteit Brussel, Belgium
Arun Das, Visa Inc., USA
Alessandro Farasin, Istituto Superiore Mario Boella (ISMB), Turin, Italy
Barbara Gili Fivela, University of Salento, Italy
Aviel Glam, Technion - Israel Institute of Technology | RAFAEL - Advanced Defence System Ltd., Israel
Josefa Gómez, University of Alcalá, Spain
Robert C. Green II, Bowling Green State University, USA
Béat Hirsbrunner, University of Fribourg, Switzerland
Fu-Hau Hsu, National Central University, Taiwan
Michael Huebner, BTU Cottbus-Senftenberg, Germany
Sergio Ilarri, University of Zaragoza, Spain
Ilias Iliadis, IBM Research - Zurich Laboratory, Switzerland
Tomayess Issa, Curtin University, Australia
Keiichi Kaneko, Tokyo University of Agriculture and Technology, Japan
Hyunbum Kim, University of North Carolina at Wilmington, USA
Ratan Lal, Kansas State University, USA
André Langer, Chemnitz University of Technology, Germany
Yiu-Wing Leung, Hong Kong Baptist University, Kowloon Tong, Hong Kong
Shigang Li, Hiroshima City University, Japan
Christopher Mansour, Mercyhurst University, Erie, USA
Alfonso Mateos Caballero, Universidad Politécnica de Madrid, Spain
Vinod Muthusamy, IBM T.J. Watson Research Center, USA
Hidemoto Nakada, AIST, Japan
Isabela Neves Ferraz, Universidade de Brasília, Brazil
Isabel Novo Corti, University of A Coruña, Spain
Petra Perner, Institute of Computer Vision and applied Computer Sciences Ibal, Germany
Xose Picatoste, University of A Coruña, Spain
Jim Prentzas, Democritus University of Thrace - School of Education Sciences, Greece
Yenumula B Reddy, Grambling State University, USA
Daniel Ritter, SAP, Germany

Claudio Rossi, Istituto Superiore Mario Boella (ISMB), Turin, Italy
Maya Sappelli, HAN University of Applied Sciences, Netherlands
Mukesh Singhal, University of California, Merced, USA
Dimitrios Skoutas, University of the Aegean, Greece
Young-Joo Suh, POSTECH, Korea
Abdelhamid Tayebi, University of Alcalá, Spain
David Tormey, Institute of Technology Sligo, Ireland
Yuehua Wang, Texas A&M University-Commerce, USA
John Woodward, Queen Mary University of London, UK
Ning Wu, School of Computer Science and Engineering - Beihang University, China
Wen-Chi Yang, NeuHelium Co. Ltd., Shanghai, China
Aleš Zamuda, University of Maribor, Slovenia
Ye Zhu, Cleveland State University, USA
Jason Zurawski, Lawrence Berkeley National Laboratory / Energy Sciences Network, USA

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Prediction of Diabetic Retinopathy and Classifiers Sensitivity Analysis <i>Khaled Almustafa</i>	1
Extraction and Use of Geometry Data to Obtain 3D Buildings on a Web Map <i>Josefa Gomez, Abdelhamid Tayebi, and Juan Casado</i>	8

Prediction of Diabetic Retinopathy and Classifiers Sensitivity Analysis

Prediction of Diabetic Retinopathy

Khaled Mohamad Almustafa

Department of Information Systems
College of Computer and Information Systems,
Prince Sultan University, Riyadh, K.S.A.
kalmustafa@psu.edu.sa

Abstract— Many eye diseases, such as Diabetic Retinopathy (DR), can lead to blindness without early clinical diagnosis, and it is extremely important to take the necessary measures before it is too late. A reliable system to detect such a disease in an early stage would be a great addition to the health care providers. In this paper, a comparative analysis of different classifiers was done for the classification of the DR dataset using different machine learning classification algorithms, such as, Naïve Bayes, J48, Random Forest (RF), Stochastic Gradient Decent (SGD), Logistic Regression (LR), Multilayer Perceptron (MP), Simple Logistic (SL) and Logistic Model Tree (LMT) classifiers, and to measure the classification accuracy, the Area Under Curve (ROC), Mean Absolute Error (MAE) and Square Root Mean Square Error (RMSE) for classifying the DR dataset. The results showed that the Logistic Regression classifier outperformed all other classifiers in the classification of the DR dataset for a classification accuracy of 74.8914%, area under curve ROC = 0.831, and RMSE = 0.4061. Then a sensitivity analysis for MP classifier was investigated in term of changing its learning rate. Also, a feature extraction method was performed on LR, MP, SL and LMT classifiers to evaluate the classification performance after selecting the relevant attributes, and the results showed that an accuracy of 72.3719% can be obtained to predict a DR case using Multilayer Perceptron by only applying a combination of up to 8 attributes instead of 19 attributes of the full dataset.

Keywords- Diabetic Retinopathy; Stochastic Gradient Decent; Logistic Regression; Multilayer Perceptron; Classification; Prediction; Feature Extraction; Sensitivity Analysis.

I. INTRODUCTION

Diabetic Retinopathy (DR), along with other eye's diseases, can lead to blindness without early clinical diagnosis. DR has a preclinical phase that can not be observed by potential patients, and such phase would be extremely important to take the necessary measures before it is too late. A reliable system to help health care providers to detect such a disease in an early stage would be a great addition to the available tools for the health care providers. Many researchers applied number of data analytics tools, as the model proposed in our paper, to classify or predict similar diseases, and to help doctors to identify these diseases on early stages. DR dataset has its own fair share from data analytics, and some of their work in this regard is presented in what follows.

Authors in [1] presented a screening of DR dataset using computer aided tools, and a reliable automatic screening system was proposed in [2][3]. Regression of DR dataset was shown in [4], and a computerized DR analysis was developed in [5], where pattern recognition techniques were presented in [6]. Authors in [7] used feature selections techniques to classify retina images, and analysis of screening systems was presented in [8]. Reliable detection techniques were presented in [9]-[12], and the role of bright lesions for DR grading with positive outcomes was investigated in [13], where retina image-level recognition was presented in [14]. A novel proposed screening algorithm was presented in [15], which was an extended algorithm with added pre-screening techniques proposed in [16]. Feature extraction was also suggested in [17], and minimizing energy cost between DR images was proposed in [18]. Machine learning classification algorithms to classify DR dataset using different classifiers were presented in [19]-[23], and deep learning was also used to classify DR dataset [24]-[26]. Decision making for automatic screening was proposed in [27]-[29], and prediction methods were applied in [30][31], and segmentation was applied in [32] to produce an efficient framework, and K-mean clustering was used in [33], and segmentation of none layers boundary for DR images was presented in [34]. Early detection of DR using deep learning for classification was proposed in [35], an ensemble based machine learning model for DR classification was presented in [36], and an automated analysis of retinal images for detection was presented in [37].

In this paper, a comparative analysis for the classification of the DR dataset using different machine learning classification algorithms, such as, Naïve Bayes, J48, Random Forest (RF), Stochastic Gradient Decent (SGD), Logistic Regression (LR), Multilayer Perceptron (MP), Simple Logistic (SL) and Logistic Model Tree (LMT) classifiers, were used to measure the classification accuracy, the Area Under Curve (ROC), Mean Absolute Error (MAE) and Square Root Mean Square Error (RMSE) for classifying the mentioned dataset. A sensitivity analysis for Multilayer Perceptron classifier was investigated to study the change of performance of the classifier in term of changing its Learning Rate parameter. Last, a feature extraction method was performed using Classifier Subset Evaluator on some

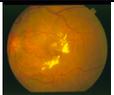
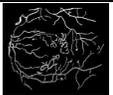
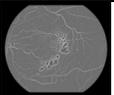
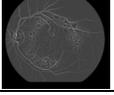
classifiers, such as LR, MP, SL and LMT, to measure the quality of the generated subsets in order to evaluate the classification performance after selecting the relevant attributes per selected classification algorithm. More details of these statistical tools used can be found in [38]. The importance of this study is to find the most suitable classifier to classify the DR dataset to help healthcare providers in early diagnosis of the DR cases, and to provide a subset DR cases' prediction based on the selected features per classifier.

This paper is organized as follows. Section 2 contains Introduction and Preparation of the DR Dataset, Section 3 introduces classification methods, Section 4 discusses methodology, and results and discussion are presented in Section 5, and Section 6 presents conclusion and future work.

II. INTRODUCTION AND PREPARATION OF THE DIABETIC RETINOPATHY DATASET

Retina images can be found in the literature and Table 1 shows some samples of these images, where the raw images are shown in the second column, and the Vessel and MSF Vessels are shown in the third and fourth columns respectively.

TABLE I. SAMPLE OF RETINA IMAGES FROM THE STARE DATASET

Sample Image	Raw Image	Vessels	MSF Vessels
1			
2			

This DR dataset used in this paper contains features extracted from the Messidor image set presented in Table 2, and this dataset is used to predict whether an image contains signs of diabetic retinopathy or not as examples seen in Table 1. The original dataset contains 1052 samples and 20 attributes (features), including the class attribute, and 611 cases with DR and 540 healthy samples. All presented features are converted from the mentioned dataset and can be seen in [3][39][40].

Table 2 contains the attributes and their given values, based on the preformed test. If we take the first row as an example, the binary values of the first attribute, denoted by 0 and 1, indicate if the image has a bad or sufficient quality, and if we take the second row as another example, the binary values of this attribute, denoted by 0 and 1, indicate if the image of the retina lacks or has a retinal abnormality,

TABLE II. ATTRIBUTE INFORMATION

Number	Attribute	Value	Note	Clarification
0	quality assessment.	0,1	0 = bad quality	Binary values

			1=sufficient quality	
1	pre-screening	0,1	1 = indicates severe retinal abnormality 0 = its lack	Binary values
2-7	MA detection.	levels alpha = 0.5, . . . , 1	Each feature value stand for the number of MAs found at the confidence	Discreet values Microaneurysm detection in retinal images
8-15	Exudates detection.	levels alpha = 0.5, . . . , 1	set of points	Discreet values
16	The Euclidean distance	0.367-0.592	of the center of the macula and the center of the optic disc to provide information regarding the patient's condition.	Continuous values
17	Diameter	0-3.087	The diameter of the optic disc.	Continuous values
18	AM/FM	0,1	AM/FM-based classification.	Binary values amplitude-modulation and frequency-modulation (AM-FM) methods for discriminating between normal and pathological retinal images.
19	Class	1 = contains signs of DR 0 = no signs of DR.	Accumulative label for the Messidor classes 1, 2, 3	Binary values

III. PROPOSED CLASSIFICATION ALGORITHMS

A number of Machine Learning classifications algorithms will be used in our analysis, in which they will be used for

model performance comparison for different classifications algorithms to classify the DR dataset.

A. Logistic Regression

We can consider the following logistic model to explain this algorithm and to see how the coefficient can be estimated from data. Given two predictors in the model x_1 and x_2 and a binary class Y , then a linear relation between x_1 and x_2 and the log-odd of the response of Y , p (response of Y), is given by:

$$\log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{1}$$

and this method is used to estimates β_i that can be used for prediction of the true and false values. Such model performs best when data separation is available in term of the positive and negative values of the data set elements.

B. Naive Bayes

Given the Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

For a given elements A and B and their probability of occurrence $P(X)$ is calculated, and such a theorem will be used to perform the classification. So for independent features, the mentioned theorem would perform a direct multiplication of the probability of each feature happening.

C. Decision Tree (DT)

A decision tree model is a model that runs number of comparison questions to divide the dataset into different smaller sets based on a given questions (Boolean for instance), and it keeps repeating the task with different set of questions for different level of the available subsets until it covers all available attributes in the dataset. We can have different type of decision tree classifiers based on the nature of the provided questions and their decision rules and based on the nature of the data set.

- a. Decision tree J48 is a special case and it is used for a unified variable associated with the dataset.
- b. Logistic Model Tree (LMT): which are classification trees with logistic regression functions at the leaves.

D. Random Forest (RF)

Random forest classifier is a collection of multiple random trees classifiers and usually an average of all trees classification results will be combined to give the performance of the random forest classification. Randomness is introduced to these trees in two different aspects:

- a. Random number of rows for each tree containing the original dataset element
- b. Random number of columns, or decision branches, for each tree

E. Stochastic Gradient Descent (SGD)

Gradient descent is an algorithm that optimizes many loss functions, such as Support Vector Machine (SVM), and Logistic Regression models, and is usually used to optimize the linear function, and the stochastic concept is introduced here based on the roots finding nature of the optimization task.

F. Multilayer Perceptron (MP)

A class of feedforward artificial neural network (ANN), and it utilizes a supervised learning technique called backpropagation for training for instances classification.

G. Simple Logistics (SL)

Simple Logistics (SL) is a binary classification model with logistic transfer function of the conditional probability of the realizations of the output variable, which is assumed to have linear combination of the input variables.

IV. METHODOLOGY

In this paper, different mentioned classification algorithms were used to compare these classifiers' performance in term of the classification of the mentioned DR dataset. For each classifier, a detailed results will be presented to compare these classifiers in terms of their classification accuracy, MAE, RMSE and ROC to select the best classifier that can be used to classify the DR dataset. Then a feature extraction method was performed using Classifier Subset Evaluator to measure the quality of the generated subsets in order to evaluate the classification performance after selecting the relevant attributes per classification algorithm, where a partial set of the full attributes will be selected for the prediction of the DR cases instead of using the full features of the original dataset. Figure 1 shows the workflow for the two used methods.

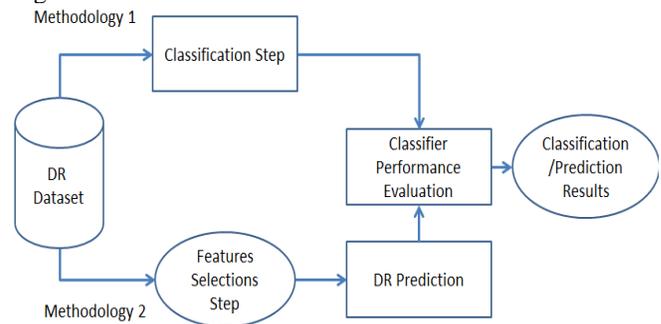


Figure 1. Proposed Methodology

V. RESULTS AND DISCUSSION

This section shows the results of the performance of the classification of the DR dataset using different classifiers, as mentioned earlier, as well as the performance of the MP classifier with some tuned parameter, mainly its learning rate, and at the end, a feature selection method on the available dataset was applied for DR prediction.

A. Using Different Classifiers

The following section describes the results obtained using different classifiers on the DR dataset with the cross validation method with 10 folds:

TABLE III. DIFFERENT CLASSIFIERS RESULTS

Classifier Used	Accuracy %	ROC	MAE	RMSE	Time (S)
NaiveBayes	63.3362	0.67	0.386	0.5356	0.04
J48	64.3788	0.68	0.379	0.5125	0.04
Random Forest	69.1573	0.76	0.390	0.4427	0.42
SGD	69.0704	0.69	0.309	0.5561	0.06
Logistic Regression	74.8914	0.83	0.323	0.4061	0.14
Multilayer Perceptron	72.0243	0.79	0.329	0.4353	2.43
Simple Logistic	71.1555	0.78	0.383	0.4313	0.64
Tree. LMT	72.1981	0.79	0.35	0.4295	3.35

The results seen in Table 3 indicates that the Logistic Regression classifier outperformed all other classifiers in the classification of the DR dataset for a classification accuracy of 74.8914%, area under curve ROC = 0.831, and RMSE = 0.4061, and it can be seen that the SGD classifier gives the best MAE results of an error value of 0.3093. Visual representation of the mentioned results of Table 3 is shown in Figure 2 and Figure 3.

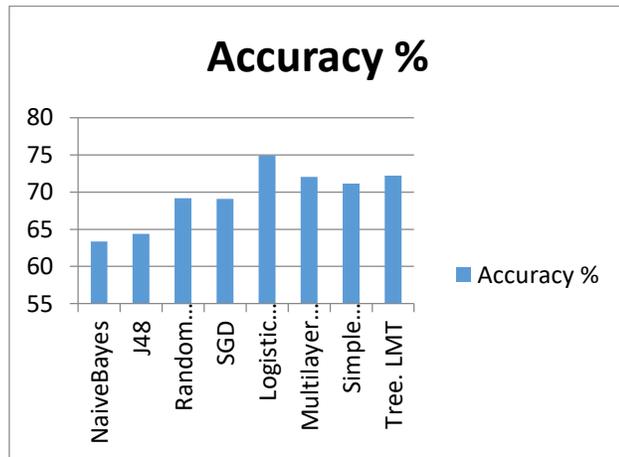


Figure 2. Classification Results in term of Accuracy for Different Classifiers

Figure 2 shows a visual representation of the classification accuracy in term of the mentioned classifiers, and Figure 3 shows a comparison results for the ROC, MAE and RMSE for different classifiers algorithms used for the classification of the DR dataset,

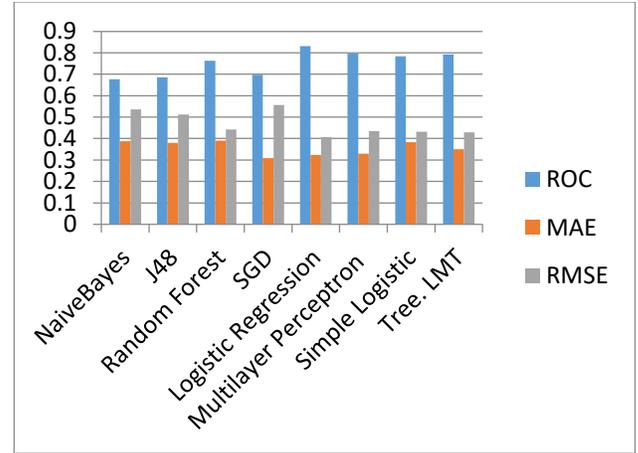


Figure 3. Classification Results in term of the ROC, MAE and RMSE Values for Different Classifiers

B. Parameter Sensitivity for some Classifiers

Parameters sensitivity for Multilayer Perceptron classifier is presented in term of changing one of its parameters, mainly the classifier Learning Rate (LR) to investigate the changes of the classifier performance, due to these changes.

1) Multilayer Perceptron Learning Rate (LR)

Learning Rate is the rate associated with the MP classifier in term of its classification weight updates, and it is a configurable parameter that influences the convergence of the algorithm:

TABLE IV. SENSITIVITY ANALYSIS OF THE MP CLASSIFIER WITH RESPECT TO LEARNING RATE

LR	Accuracy %	ROC	MAE	RMSE
0.3	72.0243	0.797	0.3298	0.4353
0.02	72.1739	0.793	0.3446	0.4302
0.01	73.4144	0.809	0.36	0.4192
0.009	70.1449	0.771	0.3775	0.4388
0.007	68.1159	0.755	0.391	0.4431
0.001	60.556	0.664	0.446	0.4661

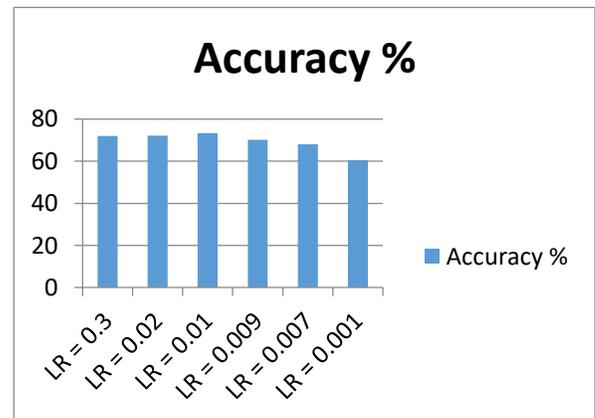


Figure 4. Classification Results in term of Accuracy for MP Classifier with LR changes

Table 4 shows the result of the performance of the MP classifier to classify the DR dataset as changes of its learning rate occurs, and we can see that the best accuracy performance is for LR = 0.01, with an accuracy of 73.4144%, ROC = 0.809 and RMSE = 0.4192, and Figure 4 shows a visual representation of the results obtained in Table 4 for the accuracy of classification with changes applied to the MP classifier learning rate.

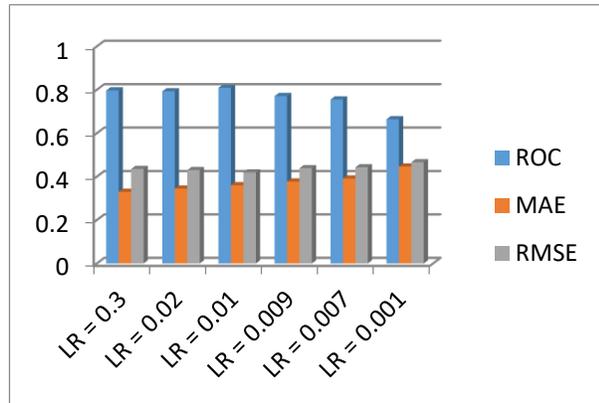


Figure 5. Classification Results in term of the ROC, MAE and RMSE Values for MP Classifier with LR changes

Figure 5 shows a comparison results for changes of the LR in term of ROC, MAE and RMSE for the MP classifier for different values of LR, and we can see that changing LR would have a small impact on the MP classifier for the classification of the DR dataset.

C. Feature Extraction

A feature extraction method was performed using Classifier Subset Evaluator by applying a training classification data to estimate the accuracy of these subsets for all used classifiers on the DR dataset, and measuring the quality of the generated subsets in order to evaluate the classification performance after selecting the relevant attributes per classification algorithm, and the results of the classifiers are shown in Table 5, and a visual representation of the results are shown in Figure 6.

TABLE V. ACCURACY RESULTS WITH FEATURE EXTRACTIONS FOR DIFFERENT CLASSIFIERS FOR HD DATASET

Features Selected	Accuracy %	Feature Selection	Selected Features (#)
Logistic Regression	74.8914	74.6308	1,2,3,4,5,6,7,8, 9,10,12,13,16,18 (14)
Multilayer Perceptron	72.0243	72.3719	2,3,5,8,9,11, 15,18 (8)
Simple Logistic	71.1555	70.808	1,3,5,8,9,10,14, 15,17 (9)
Tree. LMT	72.1981	72.1981	1,3,4,6,7,9,10, 11,12,13,14,17 (12)

Table 5 shows the results of the classification algorithms after applying the mentioned feature selection method, and it can be seen that an enhanced performance of increasing of the classifications accuracy for Multilayer Perceptron classifier from 72.0243% before applying feature selection to 72.3719%, and a reasonable performance for the Simple Logistics classifier after feature selection from an accuracy of 71.1555% before feature selection to 70.808% after feature selection. LMT classifier on the other hand showed same performance for an accuracy of 72.1981%. Figure 6 shows a visual representation of the results obtained in Table 5.

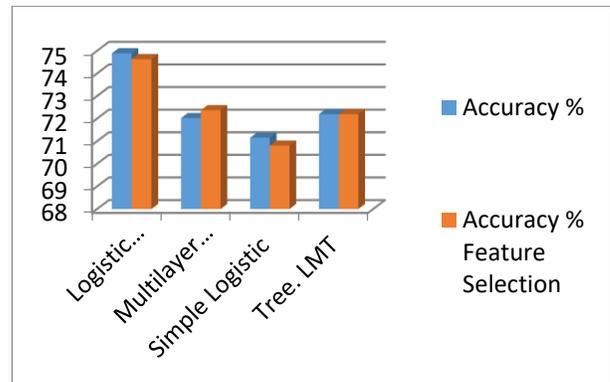


Figure 6. Visual Representation of the Results in Table 5

Table 6 shows the most relevant attributes that can be used for high accuracy classification for Multilayer Perceptron and Simple Regression classifiers, in which a reasonable accuracy of 72.3719% can be obtain to predict a DR case by only applying a combination of up to 8 attributes; mainly few MA and Exudates detections with different alpha values, and AM/FM value instead of 20 attributes of the full dataset.

TABLE VI. EXTRACTED FEATURE PER BEST PREFORMED CLASSIFIERS

Feature Number	Attribute	Note
2	MA detection.	Alpha=0.5
3	MA detection.	Alpha=0.6
5	MA detection.	Alpha=0.9
8	Exudates detection.	Alpha=0.5
9	Exudates detection.	Alpha=0.6
11	Exudates detection.	Alpha=0.8
15	Exudates detection.	Alpha=1.2
18	AM/FM	-

VI. CONCLUSION AND FUTURE WORK

In this paper, a comparative analysis of different classifiers was done for the classification of the DR dataset using different machine learning classification algorithms, such as, Naïve Bayes, J48, Random Forest (RF), Stochastic Gradient Decent (SGD), Logistic Regression (LR), Multilayer Perceptron (MP), Simple Logistic (SL) and Logistic Model

Tree (LMT) classifiers, were applied to measure the classification accuracy, the Area Under Curve (ROC), Mean Absolute Error (MAE) and Square Root Mean Square Error (RMSE) for classifying the DR dataset, The results showed that the Logistic Regression classifier outperformed all other classifiers in the classification of the DR dataset for a classification accuracy of 74.8914%, area under curve ROC = 0.831, and RMSE = 0.4061. Then a sensitivity analysis for MP classifier was investigated in term of changing its learning rate for the best performance for LR = 0.01, with an accuracy of 73.4144%, ROC = 0.809 and RMSE = 0.4192. At last, a feature extraction method was performed on LR, MP, SL and LMT classifiers to evaluate the classification performance after selecting the relevant attributes per selected classification algorithm, and a reasonable accuracy of 72.3719% can be obtain to predict a DR case using Multilayer Perceptron by only applying a combination of up to 8 attributes instead of 20 attributes of the full dataset attributes. As an extension to this work, different types of classifiers can be included in the analysis, and more in depth sensitivity analysis can be performed on these classifiers, also an extension can be made by applying same analysis to other bioinformatics dataset and see the performance of these classifiers to classify these datasets.

ACKNOWLEDGMENT

The Author would like to thank Prince Sultan University, Riyadh, KSA, and the Artificial and data Analytics (AIDA) Lab for supporting this project.

REFERENCES

- [1] M. Abramoff, M. Niemeijer, M. Suttorp-Schulten, M.A. Viergever, S.R. Russel, and B. van Ginneken, "Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes", *Diabetes Care* 31 193–198. 2008.
- [2] A.D. Fleming, S. Philip, K.A. Goatman, G.J. Prescott, P.F. Sharp, and J.A. Olson, "The evidence for automated grading in diabetic retinopathy screening", *Curr. Diabetes Rev.* 7 (4), 246–252. 2011.
- [3] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy", *Knowledge-Based Systems* 60, 20-27, 2014.
- [4] M. Abramoff, J. Reinhardt, S. Russell, J. Folk, V. Mahajan, M. Niemeijer, and G. Quellec, "Automated early detection of diabetic retinopathy", *Ophthalmology* 117(6), 1147–1154. 2010.
- [5] M. M. Fraz, P. Remagnino, A. Hoppea, B. Uyyanonvarab, A. R. Rudnickac, C. G. Owenc, and S. A. Barmana, "Blood vessel segmentation methodologies in retinal images_A survey", *Comput. Methods Programs Biomed.*, vol. 108, no. 1, pp. 407_433, 2012.
- [6] S. Soomro, F. Akram, A. Munir, C. Ha Lee, and K. N. Choi, "Segmentation of left and right ventricles in cardiac MRI using active contours", *Comput. Math. Methods Med.*, Art. no. 8350680, 2017.
- [7] C. Agurto, E.S. Barriga, V. Murray, S. Nemeth, R. Crammer, W. Bauman, G. Zamora, M.S. Pattichis, and P. Soliz, "Automatic detection of diabetic retinopathy and age-related macular degeneration in digital fundus images", *Invest. Ophthalmol. Vis. Sci.* 52 (8), 5862–5871, 2011.
- [8] M. Abramoff, M. Garvin, and M. Sonka, "Retinal imaging and image analysis", *IEEE Rev. Biomed. Eng.* 3,169–208, 2010.
- [9] B. Antal and A. Hajdu, "A prefiltering approach for an automatic screening system", in: *Proceedings of the IEEE International Symposium on Intelligent Signal Processing*, pp. 265–268, 2009.
- [10] B. Antal and A. Hajdu, "An ensemble-based system for microaneurysm detection and diabetic retinopathy grading", *IEEE Trans. Biomed. Eng.* 59,1720–1726, 2012.
- [11] H.J. Jelinek, M.J. Cree, D. Worsley, A. Luckie, and P. Nixon, "An automated micro aneurysm detector as a tool for identification of diabetic retinopathy in rural optometric practice", *Clinical Exp. Optometry* 89 (5), 299–305, 2006.
- [12] M. Niemeijer, M.D. Abramoff, and B. van Ginneken, "Information fusion for diabetic retinopathy cad in digital color fundus photographs", *IEEE Trans. Med. Imaging* 28 (5), 775–785. <<http://dx.doi.org/10.1109/TMI.2008.2012029>>. 2009.
- [13] A.D. Fleming, K.A. Goatman, S. Philip, G.J. Williams, G.J. Prescott, G.S. Scotland, P. McNamee, G.P. Leese, W.N. Wykes, P.F. Sharp, and J.A. Olson, "The role of haemorrhage and exudate detection in automated grading of diabetic retinopathy", *Br. J. Ophthalmol.* 94(6), 706–711. 2010.
- [14] C. Agurto, V. Murray, E. Barriga, S. Murillo, M. Pattichis, H. Davis, S. Russell, M. Abramoff, and P. Soliz, "Multiscale AM-FM methods for diabetic retinopathy lesion detection", *IEEE Trans. Med. Imaging* 29 (2), 502–512. 2010.
- [15] B. Antal, A. Hajdu, Z. Szab-Maros, Z. Trk, A. Csutak, and T. Pet, "A two-phase decision support framework for the automatic screening of digital fundus images", *J. Comput. Sci.* 3, 262–268, 2012.
- [16] B. Antal and A. Hajdu, "An ensemble approach to improve microaneurysm candidate extraction", *Communications in Computer and Information Science*, vol. 222, Springer-Verlag, pp. 378–394, (Chapter Signal Processing and Multimedia Applications), 2012.
- [17] X. Xu, W. Ding, M. D. Abrámoff, and R. Cao, "An improved arteriovenous classification method for the early diagnostics of various diseases in retinal image," *Comput. Methods Programs Biomed.*, vol. 141, pp. 3_9, 2017.
- [18] T. A. Soomro, T. M. Khan, M. A. U. Khan, J. Gao, M. Paul, and L. Zheng, "Impact of ICA-based image enhancement technique on retinal blood vessels segmentation", *IEEE Access*, vol. 6, pp. 3524_3538, 2018.
- [19] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis", *Med. Image Anal.*, vol. 42, pp. 60_88, 2017.
- [20] T. A. Soomro, A. J. Afifi, L. Zheng, S. Soomro, J. Gao, O. Hellwich, and M. Paul, "Deep Learning Models for Retinal Blood Vessels Segmentation: A Review", *IEEE Access* 7(1), pp:71696 – 71717, DOI:10.1109/ ACCESS.2019.2920616, 2019.
- [21] S. Doan, N. Collier, H. Xu, P. Duy, and T. Phuong, "Recognition of medication information from discharge summaries using ensembles of classifiers", *BMC Med. Inf. Decis. Making* 12 (1), 1–10. <http://dx.doi.org/10.1186/1472-6947-12-36>, 2012.
- [22] B. Antal and A. Hajdu, "A stochastic approach to improve macula detection in retinal images", *Acta Cybernetica* 20, 5–15, 2011.
- [23] L.I. Kuncheva, *Combining Pattern Classifiers, Methods and Algorithms*, Wiley, 2004.
- [24] M. Gandhi and R. Dhanasekaran, "iagnosis of diabetic retinopathy using morphological process and SVM classifier", 2013 International Conference on Communication and Signal Processing, Melmaruvathur, pp. 873-877, 2013.

- [25] D. Maji, A. Santara, P. Mitra, and D. Sheet, "Ensemble of Deep Convolutional Neural Networks for Learning to Detect Retinal Vessels in Fundus Images", arXiv:1603.04833v1 [cs.LG], 2016.
- [26] J. H. Tan1, U. R. Acharya, S. V. Bhandary, K. Ch. Chua1, and S. Sivaprasad, "Segmentation of optic disc, fovea and retinal vasculature using a single convolutional neural network", *Journal of Computational Science*, 2017
- [27] B. Antal and A. Hajdu, "Improving microaneurysm detection using an optimally selected subset of candidate extractors and preprocessing methods", *Pattern Recognit.* 45 (1), 264–270, 2012.
- [28] A.D. Fleming, K.A. Goatman, S. Philip, G.J. Prescott, P.F. Sharp, and J.A. Olson, "Automated grading for diabetic retinopathy: a large-scale audit using arbitration by clinical experts", *British J. Ophthalmol.* 94 (12), 1606– 1610, 2010.
- [29] G. Kovcs and A. Hajdu, "Extraction of vascular system in retina images using averaged one-dependence estimators and orientation estimation in hidden markov random fields", in: *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 693–696, 2011.
- [30] H. Moon, H. Ahn, R.L. Kodell, S. Baek, C. J. Lin, and J. J. Chenm, "Ensemble methods for classification of patients for personalized medicine with high-dimensional data", *Artif. Intell. Med.* 41, 197-201, 2007.
- [31] J. H. Eom, S. C. Kim, and B. T. Zhang, "Aptacdss-e: a classifier ensemble-based clinical decision support system for cardiovascular disease level prediction", *Expert Syst. Appl.* 34 (4), 2465–2479. <http://www.sciencedirect.com / science/article/pii/S095741740700139X>, 2008.
- [32] Y. Jiang, H. Zhang, N. Tan, and Li Chen, "Automatic Retinal Blood Vessel Segmentation Based on Fully Convolutional Neural Networks", *Symmetry* 2019, 11, 1112; doi:10.3390/sym11091112, 2019.
- [33] V. K. Badyal and E. S Kaur; "A New Era For Retinal Blood Vessel Segmentation Using Supervised & Unsupervised Learning Method", *International Journal Of Technology And Computing (IJTC)*, ISSN-2455-099X, Volume 2, Issue 7, 2016.
- [34] L. Fang, D. Cunefare, Ch. Wang, R. H. Guymer, Sh. Li, and S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search", Vol. 8, No. 5, 1, *Biomedical Optics Express* 2732, 2017.
- [35] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, In-Ho Ra, and M. Alazab. "Early detection of diabetic retinopathy using PCA-firefly based deep learning model", *Electronics* 9, no. 2 , 274, 2020.
- [36] T. R. Gadekallu, S. Bhattacharya, S. S. Ramakrishnan, Ch. L. Chowdhary, S. Hakak, R. Kaluri, and M. P. K. Reddy, "ensemble based machine learning model for diabetic retinopathy classification", In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1-6. IEEE, 2020.
- [37] G. Varun, L. Peng, M. Coram, M. C. Stumpe, and D. Wu, A. Narayanaswamy, S. Venugopalan, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs", *Jama* 316, no. 22, 2402-2410, 2016.
- [38] K. M. Almustafa, "Prediction of Heart Disease and Classifiers' Sensitivity Analysis", *BMC Bioinformatics*, 21, Article number: 278, ISSN: 1471-2105, 2020.
- [39] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy", *Knowledge-Based Systems* 60, 20-27, 2014.
- [40] <http://archive.ics.uci.edu/ml/datasets/Diabetic%2BRetinopathy%2BDebreceen%2BData%2BSet>

Extraction and Use of Geometry Data to Obtain 3D Buildings on a Web Map

Josefa Gómez, Abdelhamid Tayebi, Juan Casado

Computer Science Department,

University of Alcalá

Alcalá de Henares, Spain

email: josefa.gomez@uah.es, hamid.tayebi@uah.es, juan.casado@edu.uah.es

Abstract—This work shows a comparison between two different techniques to obtain 3D buildings on a web map. The first one is based on the XYZ Tiles server of OSM Buildings and the second one is based on the Overpass servers of the collaborative project OpenStreetMap. Several simulations have been carried out to analyze their performance. Benefits and limitations of both methods are discussed.

Keywords- 3D buildings; OSM Buildings; OpenStreetMap.

I. INTRODUCTION

The need of 3D geometries that model real urban environments has been growing in the last few years for different purposes, such as radio network planning, flight simulators, 3D printing, emergency management, simulations for energy consumption, etc. Concretely, the authors are working in this topic because the 3D urban model is required to develop a web application that computes the propagation loss in urban environments [1]. Therefore, a direct application of the present work is the calculation of the coverage in microcells for radio network planning in outdoor environments.

From the point of view of electromagnetism, there are two different techniques to compute propagation loss: empirical and deterministic approaches. Empirical approaches are widely used because they are fast and do not consume lots of resources. However, they are not as accurate as desired. That is the reason why deterministic methods arose. Deterministic methods are normally based on Geometrical Theory of Diffraction / Uniform Theory of Diffraction (GTD/UTD) techniques such as ray tracing. They provide extremely accurate results because they analyze the exact 3D geometrical model of any real environment. However, deterministic approaches have the disadvantages of being time-consuming and requiring lots of computing resources: both memory and processing. Another problem that hinders the application of ray tracing is the dependence of three-dimensional maps of the place under analysis, since it is quite difficult to access this type of information nowadays. In [2], authors propose to use satellite images from Google to recreate 3D buildings. Currently, other options are available like OpenStreetMap database, a collaborative project to create editable and open source maps, launched in June 2004. Everyone can easily contribute by adding new roads, buildings, points of interest like shops, churches, parks, hospitals, etc. A recent study revealed that over 1.2 million nodes and over 130.000 paths are added every day [3]. Especially in urban regions, a lot of 3D building data have been added in the last few years. In [4], OpenStreetMap data

were found out to be accurate to model the radio propagation channel for mobile communication systems. Currently, that is possible mainly due to the fact that OpenStreetMap (OSM) Buildings extract OpenStreetMap's tridimensional building data making it accessible but separated from other OpenStreetMap's data. This information can be gathered with Hypertext Transfer Protocol Secure (HTTPS) queries to a REpresentational State Transfer (REST) Application Programming Interface (API) that will provide a JavaScript Object Notation (JSON) formatted highly detailed geometry description of each building in the desired area. This has been possible thanks to OpenStreetMap flexibility that allows users from all around the world to update and modify every detail of their database easily at any point in time.

In [5], it is shown that OpenStreetMap maps are an alternative comparable with three-dimensional maps that include more details for the calculation of the propagation loss by ray tracing. The distribution of signal intensity over both maps greatly looks alike except in high density areas with low signal intensity. It has also been demonstrated that building height only represents a secondary role as long as it is established high enough. [6] shows the development of a localization algorithm based on the fingerprinting technique in which the environment has been modeled with OpenStreetMap's data. In [7], the communication channel is characterized in an urban environment through ray tracing simulations using OpenStreetMap to reconstruct the 3D model of the environment under analysis. That task was carried out in [8] by using OpenStreetMap and Flickr's data.

In this paper, a comparison between two map services that allow 3D city model generation is studied. It is worthwhile to mention that most of the tools that generate and visualize 3D city models (OSM-3D, OSM Buildings, Glosm, OSM2World, etc.) are based on OpenStreetMap. OpenStreetMap contains global building data (along with many other information). Requests can be made directly to them through Overpass. Most of the Overpass servers have limitations on the number of requests that can be done. The Main Overpass API Instance recommends not to exceed 1000 queries per day and not to download more than 5GB of data per day. Nevertheless, other Overpass servers like Kumi Systems Overpass API are more flexible and do not impose a rate limit. On the other hand, OSM Buildings contains these data from OpenStreetMap and possibly others. Requests can be made to extract data but there are also some limitations to do it. The limitations in this case are greater than in all the previous cases. The API for making requests is a REST API on an XYZ Tiles server, which returns data in GeoJSON (more convenient and concrete than

Overpass). Moreover, OSM Buildings prohibits the massive extraction of data, whereas OpenStreetMap does allow it.

As it can be seen, both methods have their advantages and disadvantages. However, the authors have not found similar works in the literature. A comparison between the two aforementioned methods is shown in Sections II, III and IV. Section II includes some simulations to analyze the response time of OpenStreetMap and Section III presents some simulations to analyze the response time of OSM Buildings. Section IV presents some common problems of both data extraction methods. Finally, conclusions are presented in Section V.

II. EXPERIMENTAL RESULTS USING OPENSTREETMAP

The Overpass API is a read-only API that serves up custom selected parts of the OpenStreetMap map data. It acts as a database over the web: the client sends a query to the API and gets back the data set that corresponds to the query.

Unlike the main API of OpenStreetMap, which is optimized for editing, Overpass API is optimized for data consumers that need a few elements within a glimpse or up to around 10 million elements in some minutes, both selected by search criteria like, e.g., location, type of objects, tag properties, proximity, or combinations of them. It acts as a database backend for various services.

Multiple servers provide access to OpenStreetMap data through this API. Requests can be written in XML language or Overpass Query Language (Overpass QL). In order to use Overpass QL, the server must provide an interpreter route that transforms the query into XML on the server side.

Responses are retrieved on XML by default, but if specified they can also be obtained in JSON, CSV and other less relevant formats. It is worth noticing that JSON responses are not the same as GeoJSON responses. GeoJSON is a well-defined spatial data format based on JSON, which is more general allowing for a more flexible, less structured format. If needed conversion from JSON to GeoJSON could be easily implemented in the client side.

Some Overpass servers provide no restrictions to access OpenStreetMap data. This is a great advantage against other options like OSM Buildings but needs to be exploited carefully. Overpass API allows to specify a timeout option that provides a simple security measure against too long or too complex queries. Nevertheless, there is no protection in the API against too big responses. A response of a big enough area that is fully populated with buildings could easily exceed the maximum heap allocation or the maximum RAM available, which will slow down or crash the process that performed the query.

A valid Overpass QL request that will fetch every building inside an area could be express as:

```
[out:json][timeout:60];(way[building](Y1,X1,Y2,X2);
relation[building][type=multipolygon](Y1,X1,Y2,X2));
out;>;out qt;
```

Executing this query against two of the mentioned overpass servers, Main Overpass API Instance and Kumi Systems Overpass API from which only the first one imposes

restrictions to access the OSM data, the following average execution times are obtained. Substituting the variables X1, Y1, X2, Y2 on the previous request, the rectangular area between 58° North, 12° East and 59° North, 13° East has been queried. This area is approximately 9438 km² and contains 70683 polygons representing buildings shapes. The following statistics were obtained.

TABLE I. COMPARISON BETWEEN OVERPASS SERVERS

	Kumi Systems Overpass API	Main Overpass API Instance
Mean	11s 639ms	12s 119ms
Standard Deviation	3s 412ms	3s 343ms
Maximum	18s 355ms	18s 107ms
Minimum	8s 169ms	7s 263ms

From the presented results in Table I, no relevant performance difference between overpass servers with and without restrictions is found.

III. EXPERIMENTAL RESULTS USING OSM BUILDINGS

One of the main characteristics of OSM Buildings is the use of GeoJSON. GeoJSON is a format for encoding a variety of geographic data structures. It is an open standard format designed for representing simple geographical features, along with their non-spatial attributes. It is based on the JavaScript Object Notation (JSON). Since query outputs already are retrieved in this spatial data standard, there will be no need to perform any transformation on the client side. As long as this is the desired data format to use.

To request data to OSM Buildings, a XYZ Tile API is used. This API divides the earth surface in rectangular regions according to a zoom size. In this case, only a value of 15 as zoom size is available. This is a great interface to iterate over contiguous tiles. It is often used by web maps to access the images that create the background over which spatial data is placed. This API is used in maps like Google Maps, OpenStreetMap or Mapbox.

In an application that calculates propagation over an area using building geometries as input, it is required to precisely delimit the area in which buildings are needed. This cannot be achieved with an XYZ Tile API. This API provides fixed tiles that cover the Earth surface seen from the specified zoom level distance. The tiles are organized in a matrix like grids on a map. Each tile is identified by its zoom level and its two the indexes on the matrix. Due to its specification, if just one small portion of the desired area is inside one of the tiles, the whole tile must be requested. It is not possible to request only a portion of a tile. Filtering the tile data to leave only the data that are also inside the desired area needs to be implemented in the client side.

OSM Buildings impose strong and limiting restrictions to access their data. The characteristics of the restrictions are not clearly specified, in contrast with the transparency on Overpass server restrictions. Experimentally, the limitations found are on the number of requests that can be performed concurrently. This restricts the maximum area to retrieve

concurrently to approximately 100 tiles. Once that limit is reached there is a cool down time, of about two minutes, during no other request from the same IP can be performed.

Data has been retrieved from OSM Buildings and Overpass in the area between 38.7028° North, 9.1955° West and 38.7540° North, 9.1189° West, which is approximately 42 km². This area corresponds to 42 tiles of zoom level 15. Comparing the obtained data (see Table II), no relevant mean time differences between Overpass and OSM Buildings are observed. Nevertheless, if the results are analyzed more carefully, the following notes can be made.

TABLE II. COMPARISON BETWEEN OVERPASS AND OSM BUILDINGS

	Kumi Systems Overpass API	OSM Buildings
Mean	4s 605ms	3s 9478ms
Standard Deviation	1s 694ms	0s 391ms
Maximum	8s 953ms	4s 7209ms
Minimum	3s 212ms	3s 338ms
Geometries	28.394	25.590

OSM Buildings is a more reliable API in the sense that it presents a minimal standard deviation in the mean request time. The reason for this might be that to retrieve the whole area multiple requests are needed, as many as the number of tiles, which could mitigate deviations on single requests.

On the other hand, Overpass API provides a greater bandwidth for data extraction on a single request. If a request to OSM Buildings had not been performed concurrently, the request time would be greater by a factor of the number of tiles.

Finally, and most importantly, is that the number of extracted geometries in the whole area is different. This has two reasons. OpenStreetMap data are improved daily by users that update, correct and polish. The new buildings added to OpenStreetMap that were not there when OSM Buildings extracted their data will not be in their API until they update them. Also, OSM Buildings filters the data removing overlapping geometries that sometimes appear duplicated on OpenStreetMap.

Taking into account that the disadvantage of both APIs is the missing building height attribute, and that the mean request time for large areas is low enough and similar between them, the selected API is Kumi Systems Overpass API because it presents lower access restrictions.

IV. BUILDING HEIGHT RECONSTRUCTION

Apart from the limitations on data extraction and the performance of the APIs the most important problem that both OpenStreetMap and consequently OSM Buildings face is the absence of the building height. Buildings have different attributes that describe them. The most important is the ground shape of the building. Others are less relevant for an application that calculates propagation, like the color of the building. But some, like the material, the roof shape and

specially the building height are quite important. Unfortunately, they are not always present.

To solve this problem multiple actions can be performed. Using the levels building attribute, which denotes the number of building levels that the building has, multiplied by a factor the denotes the building level height is quite effective. The downside of this approach is that this attribute is like the height attribute often missing.

When no attributes of the building can be used to know its height, the solutions found are to use a statistical value like the average or median height of the K nearest buildings surrounding each building with a missing height value. This approach is effective, but it is not ideal if large areas lack of the height building attribute.

On the worst case, and base on the work presented in [6], the missing heights can be substituted by a large enough value without relevantly affecting the final values of the calculated propagation. As an example, Figure 1 shows buildings directly extracted from OpenStreetMap. It is noticeable that many of the buildings lack a height value, so they are presented as plane surfaces. Figure 2 shows those same buildings with their height reconstructed. Both figures have been obtained by using the free and open-source 3D computer graphics software tool Blender [9].

V. CONCLUSIONS

The two APIs presented from which to extract building shapes presented their own strengths and faults. Some of them were shared by both since at the end they use OpenStreetMap data. Overpass was the default access point to use these data but needed extra processing on the client side to transform it to a standard data format like GeoJSON. OSM Buildings, on the other hand, presented strict access limitations that almost prevented to use it.

Since the biggest downside of both APIs (the missing building height attribute) was shared by both of them, and the mean request time for large areas was low enough and similar between them, the selected API was the one with lower access restrictions: Kumi Systems Overpass API.

ACKNOWLEDGMENT

This work is supported by the program "Programa de Estimulo a la Investigación de Jóvenes Investigadores" of Vice rectorate for Research and Knowledge Transfer of the University of Alcalá and by the Comunidad de Madrid (Spain) through project CM/JIN/2019-028.

REFERENCES

- [1] A. Tayebi, J. Gomez, F. Saez de Adana, O. Gutierrez, and M. Fernandez de Sevilla, "Development of a Web-Based Simulation Tool to Estimate the Path Loss in Outdoor Environments using OpenStreetMaps [Wireless Corner]," *IEEE Antennas and Propagation Magazine*, vol. 61, no. 1, pp. 123-129, Feb. 2019.
- [2] D. He, G. Liang, I. Portilla, and T. Riesgo, "A Novel Method for Radio Propagation Simulation Based on Automatic 3D Environment Reconstruction," *6th European Conference on Antennas and Propagation*, Prague, 2012, pp. 1445-1449.

- [3] P. Neis and A. Zipf, "Analyzing the Contributor Activity of a Volunteered Geographic Information Project-The Case of OpenStreetMap," *iSPRS International Journal of Geo-Information*, vol.1, no. 2, pp. 146-165, 2012.
- [4] J. Nuckelt, D. M. Rose, T. Jansen, and T. S. Kurner, "On the Use of OpenStreetMap Data for V2X Channel Modeling in Urban Scenarios," *7th European Conference on Antennas and Propagation*, Gothenburg, 2013, pp. 3984-3988.
- [5] T. Hänel, M. Schwamborn, A. Bothe, and N. Aschenbruck, "On the map accuracy required for network simulations based on ray launching," *16th International Symposium on World of Wireless, Mobile and Multimedia Networks*, Boston, MA, 2015, pp. 1-8.
- [6] Z. Dai, R. J. Watson, and P. R. Shepherd, "A Propagation Modeling Approach to Urban Navigation", *11th European Conference on Antennas and Propagation*, 2017, pp. 1847-1851.
- [7] L. Wang, et al., "Vehicle-to-Infrastructure Channel Characterization in Urban Environment at 28 GHz", *China Communications*, vol. 16, no. 2, pp. 36-48, Feb. 2019.
- [8] H. Fan and A. Zipf, "Modelling the world in 3D from VGI/Crowdsourced data". In: Capineri, C, Haklay, M, Huang, H, Antoniou, V, Kettunen, J, Ostermann, F and Purves, R. *European Handbook of Crowdsourced Geographic Information*, pp. 435-446. London: Ubiquity Press. 2016.
- [9] Blender computer tool. <https://www.blender.org/> [retrieved: September, 2020]



Figure 1. OpenStreetMap raw building data.

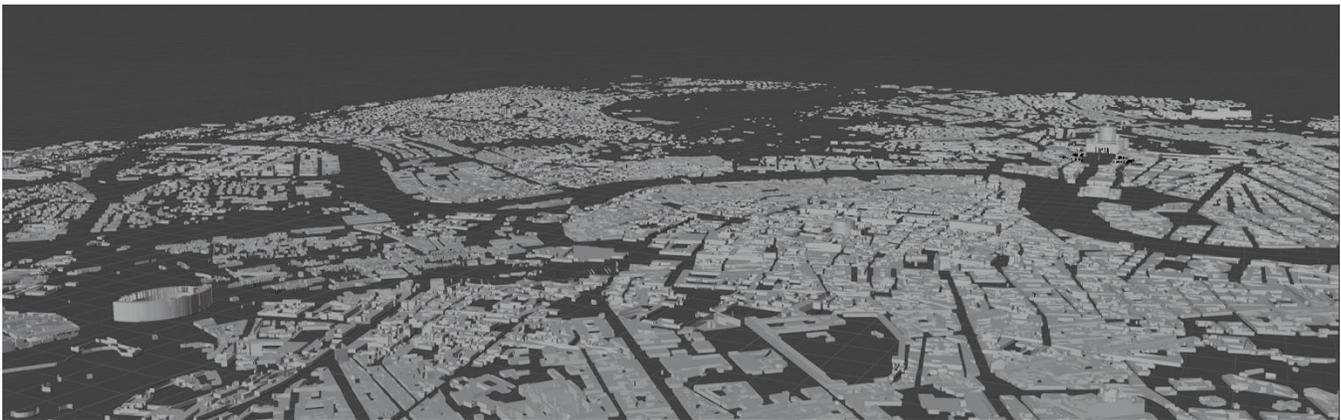


Figure 2. OpenStreetMap building data with inferred building height.