Supervised Machine Learning Algorithms for Priority Task Classification in the Cloud Computing Environment

Naoufal Er-raji[†], Faouzia Benabbou[†], Mirela Danubianu[††], Amal Zaouch[†]

[†]Laboratory of Modeling and Information Technology, Faculty of sciences Ben M'SIK, University Hassan II, Casablanca, Morocco.

[††]Stefan cel Mare University of Suceava, Faculty of Electrical Engineering and Computer Science, Suceava, Romania.

Summary

Nowadays, with the tremendous growth of request for the computing resources, the majority of the Information Technology (IT) companies start using new technologies which can give high performance resources with easy using. The cloud computing is one of the smart technologies of them. It is a new paradigm that can provide on demand services through a network (generally internet) such as servers, storage disk, platforms and applications to any Cloud Service Consumers (CSC). The CSCs focuses on minimizing response time of the service while the Cloud Service Provider (CSP) focus on efficient utilization of cloud resources in order to respect the Service Level Agreement (SLA). To satisfy both of themes, efficient methods for optimizing task scheduling have to be provided. This paper strives to use the Supervised Machine Learning Algorithms to classify the priority tasks into different tasks priority queue in order to improve the task scheduling response time.

Key words:

Cloud Computing, Data-Mining, Priority Task Classification, Supervised Machine Learning Algorithms, Task Scheduling

1. Introduction

Nowadays, the majority of the IT industry starts to use the Cloud Computing for its enormous advantages. Actually There are many definitions for the cloud computing, All of them define cloud computing as a model for enabling access to a shared pool of computing resources which can be provisioned as soon as possible and released with minimal efforts [1]. This smart technology is based on virtualization (generally it uses para-virtualization as virtualization type) and provides services to CSCs on lease basis [2]. The virtualization is the creation of one or more virtual machine in the same time and those virtual machines acting as real machines. This technology is based on Hypervisor or Virtual Machine Manager (VMM), which is a software that give the possibility to sharing resources for all virtual machines [2]. Cloud computing acts with the huge number of virtualized resources by means of scheduling. Therefore, scheduling has an important role in the cloud computing environment. In the cloud computing, CSCs can use many virtual resources for each task. Therefore, manual scheduling is not a practical solution; hence there is a need for smart task scheduling strategy. Task scheduling is the process of mapping tasks to available CSP resources on the basis of different characteristics and requirements [3-7]. Therefore, for having an efficient task scheduling, we need to use smart task scheduling algorithms. In the task scheduling practice we can use two kinds of algorithms: priority based and non-priority based [8]. Generally when an organization requires a service or resources from the CSP, a contract called SLA is signed between them. SLA is a commitment paper based on the goals and the wishes that occur between the CSPs and the CSCs. The objective of this contract is to define the rules of interaction between the CSPs and the CSCs. So if the CSC satisfied with the SLA, then requests for the cloud services [9]. In order to satisfy both the CSCs and the CSPs, a smart task scheduling strategy is needed. Recently, the use of Data-Mining for explaining the past or predicting the future makes a huge revolution in the IT decisions. Data-Mining is a multi-disciplinary ground which integrates database technology, artificial intelligence, statistics and machine learning. Data-Mining foretells the future through modeling. Modeling is the mechanism by which a model is created to foretell an outcome. If the outcome is formed as a category it is called classification and if the outcome is formed as numeric items it is called regression [10].

In this paper, we strive to use a branch of Data-Mining to classify priority tasks into different priority queue, also this paper introduce a comparison study between: Decision Tree, Linear Discriminant Analysis, K Nearest Neighbors, and Support Vector Machines. The motivation is to determine through experimental work, which method is more efficient for task classification for the priority task scheduling in the cloud computing environment. Rest of this paper is formed as follow: section 2 present some related works, section 3 will briefly describe theoretical backgrounds, section 4 presents results and discussion and finally in the last section we will present conclusion and future works.

Manuscript received November 5, 2018
Manuscript revised November 20, 2018

2. Related Work

In the literature, many researchers tried to find an efficient solution for the task scheduling either for the grid computing or the cloud computing using different procedures and mechanisms. In this section, we will present some of those solutions that have relationship with priority and Data-Mining. In [11], the proposed task scheduling strategy for the priority tasks is based on three parameters: tasks deadline, tasks age and tasks length. Also, this strategy considers that each CSP has a multi-cloud data-center, each data-center has several clusters, each cluster has several physical servers and each physical server has one or numerous virtual machines, which can give a global view on how the task scheduling are going in the cloud computing. Furthermore, this strategy gives the importance to the task classification. Kamalakar and Moulika [12] proposed an algorithm, which is based on Analytical Hierarchy Process (AHP) and multiple criteria decision making model, the proposed algorithm provides scheduling with minimum makespan, high throughput and reasonable complexity. Binodini Tripathy, Smita Dash, and Sasmita Kumari Padhy [13] proposed a solution for workflow scheduling in grid environment of multi-processors for which they are based on three novel approaches for task scheduling problem using proposed directed search optimization (DSO). Noha Hamdy, Amal Elsayed Aboutabl, Nahla ElHaggar, and Mostafa-Sami M. Mostafa [14] proposed a solution for task scheduling in the cloud computing based on artificial neural networks (ANN) optimized with firefly algorithm. The intention is to minimize the training error and realize quick convergence rate. Experiments show that CSC can specify the appropriate task-scheduling algorithm in different circumstances. V. Venkatesa Kumar and K. Dinesh [15] proposed the use of Fuzzy Neural Network for the task scheduling. Firstly, they classify the user tasks based on Quality of Service parameters. Afterwards, the results of the classified tasks are given to the fuzzier.

3. Theoretical Backgrounds

In the cloud computing environment, tremendous numbers of different tasks keep coming from the CSCs to the CSP resources and need to be executed as soon as possible with the minimum time which means having minimum response time. The response time is the time elapsed between task submission and the response the user gets from the CSP system. It includes the waiting time, task execution time as well as the tasks classification time [16]. So, task classification time within the task scheduling can have a big impact in the overall response time. In the last decades, Data-Mining makes a big revolution in the computer sciences decisions for its amazing results. In the Data-Mining, there are different approaches that can be used for different purposes.

In Data-Mining there are two approaches: descriptive and predictive. For the description there are: Deviation detection, Clustering, Association Rules, and Visualisation. And for the prediction the future there are: classification and regression.

In this paper, we are concerned with the classification therefore; we will use as a technique for the classification the following methods: Support Vector Machine, Linear Discriminant Analysis, Decision Tree and K-Nearest Neighbours.

As we said before, the Data-Mining is divided into two parts description and predicting the future. It works by the means of data analysis which produce a particular enumeration of patterns from such data. It is a multi-disciplinary ground which integrates database technology, artificial intelligence, statistics and machine learning. In the Data-Mining the future is predicted through the modeling. The modeling is the mechanism by which a model is created to foretell an outcome. If the outcome is formed as a category it is called classification otherwise if the outcome is formed as numeric items it is called regression [10]. Classification is a Data-Mining function that intent to foretell a value of a categorical variable (target or class) through building a model stand on one or more parameters (predictors or attributes).

The main algorithms used in Data-Mining for the classification are:

1. Decision Tree [17]: It constitutes a classification model through a tree structure. It splits a data-set into smaller subsets simultaneously an associated decision tree is incrementally developed. As a result a tree with decision nodes and leaf nodes is created. For each leaf node assigns a class and for each decision node tests an attribute.
2. Ensemble Classifier [18]: It is based on multiples base classifiers that are cooperatively trained on data-set in a supervised classification problem. During learning, the base classifiers are trained separately on the data-set.
3. Support Vector Machines [19]: It is based on a supervised machine learning algorithm. In this algorithm, we design each data component as a point in n-dimensional space (where n is number of features that we have) with the value of each feature, there is an association with the value of a particular coordinate. Next, we start the classification through finding the hyper-plane which differentiates the classes.
4. K Nearest Neighbors [20]: It is based on a simple algorithm that stocks all available cases and classifies new cases by means of a similarity

measure. (e.g., distance functions: Euclidean Distance, Minkowski Distance and Mahalanobis Distance).

4. Results and Discussion

Generally, the tasks can be scheduled (executed) according to different parameters (requested resources availability, requested resources load, task priority, etc.). As we will use the supervised machine learning algorithms for the classification, we have to provide a known set of input data (Tasks and their characteristics (parameters)) and known responses (Assigned Priority Queue) for the training step. Therefore, we need a set of tasks parameters that can guarantee a good task scheduling with an algorithm to manage the response value. So, we have chosen the three best QoS parameters which are: task deadline, task length and task age. The deadline is chosen as a priority parameter by reason of it is one of the most important parameter to be respected in the QOS, which is documented in the SLA. Also, we have chosen the task age because when the task has a low priority, it has to wait for a long time and this leads to increase in execution time. Finally we have chosen the task length because when the task has small length it can execute rapidly, thus liberate the resources as soon as possible for the other tasks. After the choice of the three parameters that we will base on; we have to assign a value for each task. Because we didn't found any database that contains such parameters, we were obliged to assign a random value for each one. After, this step we completed our data-set by attribution to each task a priority queue using our algorithm, which considered as a learning algorithm. The algorithm pseudo-code used for making the data-set is given in detail below.

Priority Task Classification – Data-set

Received tasks
Sort tasks according **to** their deadline
Repeat
For i **from** 1 **to** the sorted task size **do**
Find the Min_Length **&& Find** Max_Length
Find the Min_Age **&& Find** Max_Age
End For
For i **from** 1 **to** the sorted task size **do**
Switch (Ti [Tl] **&&**Ti[Ta]) {
Case Ti[Tl] = Min_Length **&&** Ti[Ta] = Max_Age
    Then put Ti **in** QP1
Case Ti[Tl] = Min_Length **&&**Ti[Ta] ≠ Min_Age **&&** Ti[Ta] ≠ Max_Age
    Then put Ti **in**QP2
Case Ti[Tl] = Min_Length **&&** Ti[Ta] = Min_Age
    Then put Ti **in** QP3
Case Ti[Tl] = Length ≠ Min_Lenght **&&** Ti[Tl] ≠ Max_Length **&&** Ti[Ta] = Max_Age
    Then put Ti **in** QP4
Case Ti[Tl] = Length ≠ Min_Lenght **&&** Ti[Tl] ≠ Max_Length **&&** Ti[Ta] ≠ Min_Age **&&** Ti[Ta] ≠ Max_Age
    Then put Ti **in** QP5
Case Ti[Tl] = Length ≠ Min_Lenght **&&** Ti[Tl] ≠ Max_Length **&&** Ti[Ta] = Min_Age
    Then put Ti **in** QP6
Case Ti[Tl] = Max_Length **&&** Ti[Ta] = Max_Age
    Then put Ti **in**QP7
Case Ti[Tl] = Max_Length **&&**Ti[Ta] ≠ Min_Age **&&** Ti[Ta] ≠ Max_Age
    Then put Ti **in**QP8
Case Ti[Tl] = Max_Length **&&** Ti[Ta] = Min_Age
    Then put Ti **in**QP9
Case Ti[Tl] **=** Min_Lenght = Max_Lenght **&&** Ti[Ta] = Min_lenght = Max_Length
    Then put Ti **in** QP10
Until all tasks are assigned

The obtained results will be used to train all the Supervised Machine Learning algorithms in order to use it with the new tasks.

In order to verify the effectiveness of our algorithms, we used the Classification Learner App. This app is one of the Matlab APP which can give the possibility to trains models to classify data. With the use of this application we can analysis supervised and unsupervised machine learning algorithms with the use of different classifiers. We can explore our data, specify validation schemes, train models, appraise results and select features. Furthermore, this application gives us the possibility to perform automated training in order to search for the finest classification model type [21].

In this paper, we will use different methods with different model type, in the following a brief definition of the used models.

Complex Tree: uses a lot of leaves which make a lot of fine distinctions between classes. For this classifier the maximum number of split is fixed on 100.

Medium Tree: uses fewer leaves than complex tree. For this classifier the maximum number of split is fixed on 20.

Simple Tree: uses less leaves which make coarse distinctions between classes. For this classifier the maximum number of split is fixed on 4.

Linear SVM (Support Vectors Machines): uses a linear kernel also it makes a simple linear separation between classes.

Quadratic SVM: uses a Quadratic Kernel.

Cubic SVM: uses a Cubic Kernel.

Fine Gaussian SVM: uses the Gaussian Kernel with Kernel Scale also it makes Finely-detailed Distinctions between classes.
Medium Gaussian SVM: uses the Gaussian Kernel with Kernel Scale. Also, it makes less distinction than a fine Gaussian SVM.
Coarse Gaussian SVM: uses the Gaussian Kernel with Kernel scale; also it makes Coarse distinctions between classes.
Fine KNN: form detailed distinctions between classes.
Medium KNN: form less distinctions compared with Fine KNN.
Coarse KNN: form coarse distinctions between classes.
Cubic KNN: uses a cubic distance metric.
Weighted KNN: uses a distance weighting.
Boosted Tree: creates an ensemble of medium decision trees with the use of Adaboost algorithm compared to bagging, boosting algorithms use relatively little time or memory, but might need more ensemble members.
Bagged Tree: is a bootstrap-aggregated ensemble of complex decision trees often very accurate, but can be slow and memory intensive for large data-set.
Subspace Discriminant: is superb for lots of predictors. It uses less memory and fast for fitting and predictions however, the accuracy differs depending on the used data. In addition, the model creates an ensemble of discriminant classifiers with the use of the random subspace algorithm.
Subspace KNN: almost is used when we have many predictors. The model creates an ensemble of nearest-neighbor classifiers using the random subspace algorithm.
Rusboosted Trees: almost is used when we have skewed data with many more observation of one class.
As we said before we will use different supervised classifiers. Thus, we applied those classifiers to different numbers of tasks: 100, 150, 200, 250, 300, 350, and 400 successively.
In this paper, we used three parameters for the comparison between the classifiers.

- Accuracy, which is a percentage that calculated as: the number of the correct prediction divided by the total number of prediction multiplied by 100 [22].
- Prediction speed, which is the time spent to obtain a model from a train data-set, for a given hardware and a specific sample size [23].
- Training time, which is taking the product of the number of epochs and the time taken by each epoch [24].

In the following, the obtained results:

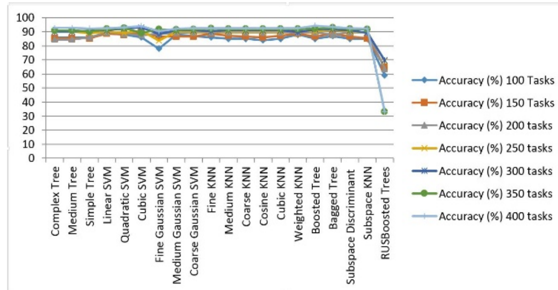


Fig. 1 Accuracy Comparison

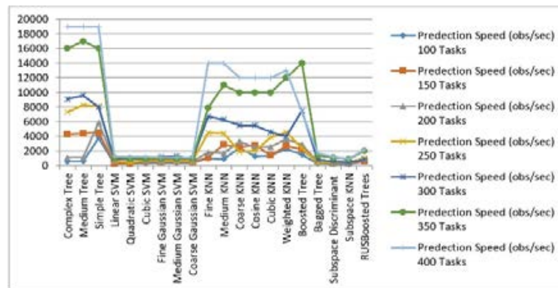


Fig. 2 Prediction Speed Comparison

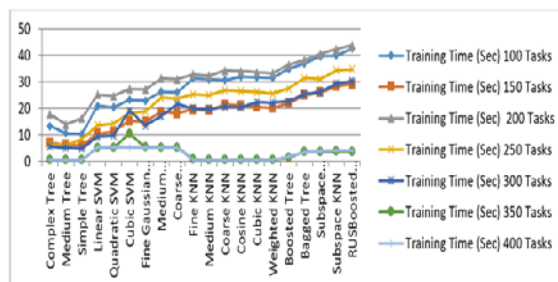


Fig. 3 Training Time Comparison

In this paper, we used different classifier models in order to classify priority tasks into different priority queues. As shown in the three Figs, we have different results depending on the tasks numbers and which model is used.
As shown in the Fig one the maximum accuracy is 94.3 % (Boosted Tree at 400 Tasks) and the minimum accuracy is 33 % (Rusboosted tree at 400 tasks). Also, all accuracy values are higher than 80 % except Fine Gaussian SVM at 100 Tasks and Rusboosted Trees at all numbers of tasks. In addition, we can see that all the accuracy is arising at 400 tasks except Fine Gaussian SVM and Rusboosted trees.
As shown in the Fig two the maximum prediction time is 1900 obs/sec (complex, medium, Simple)-Tree at 400 tasks) and the minimum prediction time is 190 obs/sec (Subspace KNN at 100 tasks). Also, all models have equal or less than 2000 (obs/sec) except: Simple Tree, (Coarse-Weighted)-KNN at 100 Tasks. (Boosted, Medium, Complex, Simple)-Tree and (Medium, Cosine, Corse,

Weighted) - KNN at 150 Tasks. (Boosted, Simple)-Tree, (Weighted, Cubic, Cosine, Coarse)-KNN, at 200 Tasks. (Simple, Medium, Complex, Boosted)-Tree, (Weighted, Medium, Cubic, Fine)-KNN, at 250 Tasks. (Boosted, Simple, Medium, Complex)-Tree, (Medium, Fine, Weighted, Cosine, Cubic, Coarse)-KNN at 300 Tasks. (Boosted, Complex, Medium, Simple)-Tree, (Fine, Medium, Coarse, Cosine, Cubic, Weighted)-KNN at 350 Tasks. (Rusboosted, Boosted, Simple, Medium, Complex)-Tree, (Weighted, Cubic, Fine, Medium, Coarse, Cosine)-KNN at 400 Tasks.

As shown in the Fig three the maximum Training Time is 43.766 sec (RUSBoosted tree at 200 Tasks) and the minimum training Time is 0.50631 sec (Simple Tree at 350 tasks). Furthermore, all models have equal or less than 5s except Simple Tree at 300 tasks, (Rusboosted, Complex, Simple, Medium, Bagged, Boosted)-Tree, Subspace Discrimant, (Subspace, Weighted, Cosine, Coarse, Medium, Fine, Cubic)-KNN, at 350 tasks, (Rusboosted, Bagged, Simple, Medium, Complex, Boosted)-Tree, Medium Gaussian SVM, Subspace Discrimant, (Subspace, Weighted, Cosine, Medium, Fine, Coarse, Cubic)-KNN at 400 tasks.

The all obtained results show that there is no perfect and no worse model for all numbers of tasks in term of three parameters. However, those results let us make a conclusion that for having a good results we have to have high accuracy with minimum prediction time and minimum training time, for that we have to use a good training algorithm with a big amount of training data. Moreover, from those results if we consider just one parameter we can found the best one. Thus, the most important for having a good task scheduling is having a good task scheduling strategy which can give a global overview of all what happened from the beginning when the tasks are coming to the cloud until the task execution in the Virtual Machines.

5. Conclusion and Future Work

Cloud computing is a recent technology that trend to help companies as well as researchers to use services in scalable manner. Therefore, the use of these services capabilities required many procedures in order to get better performance. One of the most important procedures is the task scheduling for its relationship with response time and SLA. In this paper, we used different Supervised Machine Learning Algorithms for the task classification. Furthermore, we gave a comparison between different solution in term of accuracy, prediction speed and training time. On the other hand, we have shown that those classifiers had given very interesting results 90% which allow us to validate the consistency of our algorithm as a task scheduler. As a future work, as long as our algorithm is based on many criteria and the objective is to optimize the tasks assignment according to those criteria, we propose the use of Analytic hierarchy process (AHP) to implement hierarchical classifications which exactly give the requirements of the task scheduling with priorities. Moreover, we will take into account task migration, the dependant task, and study the impact of this approach on the overall performance obtained with the respect of the SLA.

References

[1] International Organization for Standardization, Information technology—Cloud computing—Overview and vocabulary, (2014), ISO/IEC 17788. International Organization for Standardization, Information technology—Cloud computing—Overview and vocabulary, (2014), ISO/IEC 17788.

[2] Daniel A. Menascé, Virtualization: Concepts, Applications, And Performance Modeling , Dept. of Computer Science George Mason University Fairfax, VA 22030, USA.

[3] Naoufal Er-Raji, Faouzia Benabbou, Ahmed Eddaoui, Task Scheduling Algorithms in the Cloud Computing Environment: Survey and Solutions , (2016) International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 1 (6).

[4] Abdul Razaque, Nikhileshwara Reddy Vennapusa, Nisargkumar Soni, Task Scheduling in Cloud Computing, Systems, Applications and Technology Conference (LISAT), Long Island – (IEEE-2016).

[5] Varinder Kaur, Gurjot Kaur, Cloud Computing Scheduling Algorithms: A Review, (2015) International Journal of Computer Science and Network (IJCSN) 2 (4).

[6] Arnav Wadhonkar , Deepti Theng, A Survey on Different Scheduling Algorithms in Cloud Computing, International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16), (IEEE-2016).

[7] Vigindra, Sudhir Shenai, Survey on Scheduling Issue in Cloud Computing, international conference on modeling optimization and computing – (ICMOC-2012).

[8] Nidhi Bansal, Dr. Maitreyee Dutta, Performance Evaluation of Task scheduling with priority and non-priority in cloud computing, International Conference on Computational Intelligence and Computing Research – (IEEE-2014).

[9] G. Justy Mirobi , Dr. L. Arockiam, Service Level Agreement In Cloud Computing. An Overview, International Conference on Control Instrumentation, Communication and Computational Technologies (LCCICCT) – (IEEE-2015).

[10] Fabricio Voznika , Leonardo Viana, Data Mining Classification, (Springer-2001).

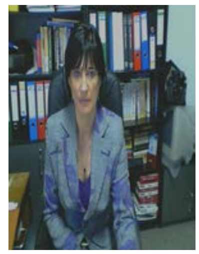
[11] Naoufal Er-raji, Faouzia Benabbou, Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing, (2017) International Journal of Advanced Computer Science and Applications (IJACSA), 2 (8).

[12] Kamalakar. M, Moulika.T, A Priority Based Job Scheduling Algorithm in Cloud Computing, (2015) International Journal Of Innovation Technology (IJITECH), 1 (3).

[13] Binodini Tripathy, Smita Dash, Sasmita Kumari Padhy, Dynamic task scheduling using a directed neural network" (2015) journal of parallel and distributed computing-Elsevier.
[14] Noha Hamdy, Amal Elsayed Aboutabl, Nahla ElHaggar, Mostafa-Sami M. Mostafa, Dynamic Task Scheduling Method in Cloud Computing Environment Using Optimized Neural Network (2017) International Journal of Future Computer and Communication (IJFCC), 4 (6)
[15] V. Venkatesa Kumar, K. Dinesh, Job Scheduling Using Fuzzy Neural Network Algorithm in Cloud Environment, (2012) Bonfring International Journal of Man Machine Interface, 1(2).
[16] Peter M. Broadwell, Response Time as a Performability Metric for Online Services, Computer Science Division University of California at Berkeley, Report No. UCB//CSD-04-1324, May 2004.
[17] Tom M. Mitchell, Machine Learning (1997) McGraw-Hill Science-Engineering-Math.
[18] Akhlaqur Rahman, Sumaira Tasnim, Ensemble Classifiers and Their Applications: A Review, (2014) International Journal of Computer Trends and Technology (IJCTT), 1(10).
[19] Support Vector Machines : https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[20] Izabela Moise, Evangelos Pournaras, Dirk Helbing, K-Nearest Neighbour Classifier, ETH Zurich.
[21] Classification Learner App: https://www.mathworks.com/help/stats/classification-learner-app.html.
[22] Elham Akhond Zadeh Noughabi, Bijan Raahemi, Amir Albadvi, Behrouz H. Far, Handbook of Research on Data Science for Effective Healthcare Practice and Administration, IGI Publishing Hershey, PA, USA- 2017.
[23] M. Clemente, V. Giner-Bosch, S. San Matías, Assessing classification methods for churn prediction by composite indicators, Department of Applied Statistics, Operations Research and Quality, Universitat Politècnica de València, Camino de Vera s/n, 46022 Spain.
[24] Yasaswi, Jitendra, Suresh Purini und C. V. Jawahar: Predicting the Training Time of Deep Neural Networks. Technischer Bericht, IIIT Hyderabad, India, 2016.

Naoufal Er-raji received his master's degree in computer science (networks and systems). He is currently a PhD Student in the university Hassan II Casablanca, Morocco. His present research interests are: task scheduling, cloud computing, big data, Data-mining.
Email: na.erraji@gmail.com

Faouzia Benabbou is a professor of Computer Science and member of Computer Science and Information Processing laboratory. She is Head of the team "Cloud Computing, Network and Systems Engineering (CCNSE)". She received his Ph.D. in Computer Science from the Faculty of Sciences, University Mohamed V, Morocco, 1997. His research areas include cloud Computing, data mining, machine learning, and Natural Language Processing. She has published several scientific articles and book chapters in these areas.
Email: Faouzia.benabbou@univh2c.ma



Mirela Danubianu (b. July 13, 1961) has obtained the B.S. and M.S. degree in Computer Science from University of Craiova in 1985, and the PhD. degree in Computer Science in 2006 from "Stefan cel Mare"University of Suceava. She has also obtained the M.S.. degree in Economics from University of Craiova in 2001. Currently, she is Associate Professor and Head of the Computers Department at "Stefan cel Mare University" of Suceava.
Her current research interests include databases theory and implementation, data mining and data warehousing, big data, and application of advanced information technology in economics and health care area.
Email: mdanub@eed.usv.ro

Amal Zaouch Born on 9 september 1983 in oujda city , morrroco Receivevd her master 's degree in (telecommunication and networks engineer) in 2006 at the National School of Applied Sciences, Oujda, Morocco. Her present research interests are: Load balancing, cloud computing, big data, data mining.
Email: zaouch_amal@yahoo.fr