IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2019, accepted February 5, 2019, date of publication February 19, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899985

Using Machine Learning Ensemble Methods to Predict Execution Time of e-Science Workflows in Heterogeneous Distributed Systems

FARRUKH NADEEM, DANIYAL ALGHAZZAWI, ABDULFATTAH MASHAT, KHALID FAQEEH, AND ABDULLAH ALMALAISE

Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Farrukh Nadeem (fabdullatif@kau.edu.sa)

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH) - King Abdulaziz City for Science and Technology - the Kingdom of Saudi Arabia - award number (12-INF2716-03). The authors also, acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support.

ABSTRACT Effective planning and optimized execution of the e-Science workflows in distributed systems, such as the Grid, need predictions of execution times of the workflows. However, predicting the execution times of e-Science workflows in heterogeneous distributed systems is a challenging job due to the complex structure of workflows, variations due to input problem-sizes, and heterogeneous and dynamic nature of the shared resources. To this end, we propose two novel workflow execution time-prediction methods based on the machine learning ensemble models. In this paper, we showcase our approach for two different real Grid environments. Our approach can effectively predict the execution time of the scientific workflow applications in the Grid for various problem sizes, Grid sites, and runtime environments. We characterized the workflow performance in the Grid using the attributes that define structure of workflow as well as the execution environment. Contrary to common ensembles, our ensemble systems employed three strong learners, which balance the weaknesses of each other by their strengths to model the workflow execution times. The proposed methods have been thoroughly evaluated for three real-world e-science workflow applications. The experimental results demonstrated that our proposed multi-model ensemble models can significantly decrease the prediction error (by 50%, on average) as compared with methods based on the radial basis function neural network, local learning, and performance templates. The proposed methods can also be applied with similar effectiveness and without any major modification for other heterogeneous distributed environments, such as the Cloud.

INDEX TERMS Machine learning ensemble systems, performance modeling of e-science workflows, distributed execution of e-science workflows, execution time prediction of e-science workflows.

I. INTRODUCTION

The Grid has emerged as a collection of heterogeneous resources from multiple domains, where different policies are observed for the availability of the resources and shared access of the resources is granted to the users. Scientific workflow applications comprise several different tasks to be executed in a specific order, referred to as scientific workflows. Scientific workflows are commonly executed in the Grid in a distributed way and managed using a workflow management system, like Pegasus [1], Askalon [2], etc. Several services in high level middleware (e.g. resource provisioning, workflow task scheduling [3], performance monitoring and analysis [4], runtime performance tuning, etc.) require predictions of workflow performance for planning, monitoring, and optimizing workflow execution in the Grid. Besides, the workflow user need this prediction to manage his submissions and the resource owners need it for capacity planning of their resources.

Predicting workflow execution time in the Grid for different Grid-sites is a hard job mainly due to dynamic nature of runtime environment, complex structures of the workflows, shared nature of Grid-resources, and several other factors (see Section II for more details). In addition, varying

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Tian.

input problem-sizes, different resource sharing policies, and user rights on different Grid-sites make it even more difficult. To address this problem, we propose two multi-model machine learning ensemble systems to predict execution time of e-science workflows in the Grid. Ensemble modeling is a powerful machine learning method that combines the power of multiple learners to improve the prediction accuracy of the overall model. To begin with an objective of fine grain modeling of workflow performance in the Grid, we characterized workflow performance in term of attributes that cover all major factors affecting workflow performance, e.g. workflow structure, application attributes, submission and execution environments, etc. The workflow execution traces (referred to as datasets) comprising these attributes were collected from two different Grid environments (see Section IV-B). To model workflow execution time from workflow attributes, we employed three different strong learners – the methods which can be used as standalone prediction methods – as ensemble members. We trained the ensemble members by dividing the trace dataset in two subsets: training dataset and test dataset. The ensemble members were joined using two methods: mixture of experts and dynamic experts selection. The mixture of experts combines the weighted predictions of ensemble members (referred to as experts) to generate the final prediction. The weights of ensembles are generated through a gating network that is trained using expectation maximization [5], [6] of accuracy of the final predictions. The dynamic experts selection evaluates the experts based on their accuracy in the vicinity of the given inputs for prediction. It then chooses the expert with highest accuracy to generate the final prediction. The vicinity is determined in terms of k-nearest neighbors, where the distance is calculated by adapting Heterogeneous Euclidean-Overlap Metric [7]. The proposed methods were evaluated for three real world scientific workflows in two different Grids. Our experiments showed that the presented approach can decrease the prediction error by 50% (on average) as compared with previous approaches.

The major contributions of current study include the followings.

- two novel multi-model machine learning ensembles methods to predict execution times of complete workflows in the Grid for varying input problem-sizes and different Grid-sites, which to the best of our knowledge have not been used before for execution time predictions of entire workflows;
- a flexible (in terms of attributes defining workflow performance in all phases during its execution) and scalable (number of strong learners can be increased to further improve the prediction accuracy) prediction approach that can be easily adjusted for the available information;
- an experimental evaluation of the proposed methods for three real-world e-science workflows in two different real Grid environments (not simulations);
- a higher accuracy of workflow execution times predictions in the Grid than any other approach in related work that incorporates all our supported static as well as dynamic workflow attributes; and
- notably, our methods can effectively be applied for other distributed environments like the Cloud, provided that suitable parameters defining workflow performance in the target environment are identified and the corresponding trace data is available.

The remaining of this paper is arranged as following. Section II describes parameterizing performance of scientific workflows in the Grid. Section III presents our multi-model ensemble systems for predicting workflow execution time. Section IV describes our experiments, the scientific workflows for our experiments and the tested environment for evaluation of the proposed methods. Related work is described in Section V. Finally, we conclude the current study and describe future work in Section VI.

II. PARAMETERIZING PERFORMANCE OF SCIENTIFIC WORKFLOW APPLICATIONS IN THE GRID

Workflow execution time can be effectively modeled using different attributes reflecting workflow structure and execution environment. Nadeem and Fahringer [8] and Nadeem *et al.* [9] presented a comprehensive framework defining attributes that can effectively model workflow execution time in the Grid. We used their framework to parameterize workflow performance in terms of workflow attributes. These attributes cover workflow static as well as runtime information. The static information primarily includes structural attributes, e.g. tasks in workflow, task dependencies, etc. The runtime information includes the attributes defining Grid runtime environment, e.g. selected scheduling algorithm [4], selected Grid-sites [10] and their states, etc. For the sake of better understanding the proposed approach, we are reproducing the detailed attributes from [8] and [9] as below.

□ Workflow structure attributes
  * Workflow name: it is the name of the scientific workflow.
  * Workflow tasks: these are executables or data transfers in the workflow.
  * Dataflow dependencies: show that data taken as an input of a task is an output of another.
  * Control flow dependencies: represent the sequence of carrying out tasks in workflow.

□ Workflow attributes
  * Executables: these are names of the softwares corresponding to the tasks in the workflow.
  * Versions: describe the type of the executables (e.g. serial/parallel version etc.).
  * Problem-size: it describes the problem given as an input for execution of the workflow.
  * File sizes: describe the size of files taken as input.

□ Submission environment attributes
  * Names of Grid-sites: the Grid-sites chosen for execution of the workflow.
  * Scheduling strategy: it is the strategy employed to assign individual tasks to chosen Grid-sites.

  * Time of submission: clock time when workflow is submitted for execution.
- □ Grid-site state attributes
  * Pre-queued jobs: describes jobs pre-submitted in the queue when the workflow task is submitted to the queue.
  * Pre-queued CPUs: represents number of CPUs requested by pre-queued jobs.
  * Pre-executing jobs: describes the jobs executing at the time of submission of workflow task.
  * Pre-allocated CPUs: represents number of CPUs being used by the pre-running jobs.
  * Parallel jobs: describes jobs executing in parallel with the workflow task.
  * parallel occupied CPUs: represents CPUs being used by the parallel jobs.
- □ Policies for sharing resources
  * User name: user name of the person executing workflow. Grid-sites usually observe different resource sharing policies for different users.
  * Virtual organization or Group: it is the name of the virtual organization or group the user belongs to.
- □ Network attributes
  * Bandwidth: represents network bandwidth available for transferring (any) workflow data.
  * Latency: describes network latency at the time of the data transfers.
  * Parallel transfers: describes number of parallel data transfers for optimized data flow.

III. MULTI-MODEL ENSEMBLE SYSTEMS FOR WORKFLOW EXECUTION TIME PREDICTIONS

In this section, we describe our novel method based on multi-model machine learning ensemble systems for prediction of workflow execution time in the Grid. The main objective of our ensemble systems is to reduce the variance in prediction – and thus improve the overall prediction accuracy – of individual prediction methods by using a combination of prediction methods. There were three main phases to build our ensemble system, each of which required selection of an effective strategy for overall effectiveness of the ensemble system. These phases are shown in the Figure 1 and are described below.

1) Ensuring diversity in output of ensemble members
2) Selection and training of ensemble members
3) Combining ensemble members

A. ENSURING DIVERSITY IN OUTPUT OF ENSEMBLE MEMBERS

In phase-1, a strategy is required to ensure diversity in the output of ensemble members (EMs) (see Section III-B). This can be done in one of the two ways: by using heterogeneous EMs or by using different data samples to train one EM. In this work, we used a set of heterogeneous EMs. The selected EMs used the prediction methods that are completely different in nature as well as in algorithms, and thus result in different prediction accuracies. Since our EMs are different from each other, we selected a large part of our trace dataset (90%) as one data sample for training of EMs (the remaining 10% was later used in evaluation of ensemble system).

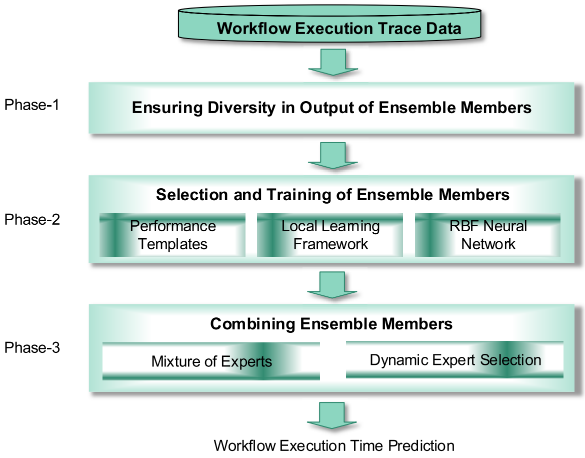

Workflow Execution Trace Data
Phase-1
Ensuring Diversity in Output of Ensemble Members
Phase-2
Selection and Training of Ensemble Members
Performance Templates
Local Learning Framework
RBF Neural Network
Phase-3
Combining Ensemble Members
Mixture of Experts
Dynamic Expert Selection
Workflow Execution Time Prediction

FIGURE 1. Different phases of building and ensemble system.

B. SELECTION AND TRAINING OF ENSEMBLE MEMBERS

In current study, we chose workflow execution time prediction methods based on local learning [11], performance templates generated through evolutionary programming [8] and radial basis function neural network [9] as our ensemble members. The reasons behind choosing these methods include heterogeneous nature of prediction algorithms of these methods and easy availability of their implementation. Above all, these methods appeared to be more effective than several other methods in related work [9]. The selected prediction methods were implemented and trained as described in their sources [8], [9], and [11], respectively. For the sake of consistency, and due to space limitation, we are not reproducing details of these methods in this paper and refer the reader to the mentioned sources for their details. In this paper, we focus on details of our ensembles for workflow execution predictions. The trace dataset used for training and evaluation of ensembles consisted of sets of workflow attribute values $X = \{x_1, x_2, \ldots, x_n\}$ and corresponding real execution time t_i, i.e. $\{(X_i, t_i)\}$. For description of trace dataset see Section IV-B. We divided trace dataset into two subsets: a large set (90% of trace dataset), referred to as training dataset, for model building of ensembles and a small set (10% of trace dataset), referred to as test dataset, for testing and evaluating the ensemble system.

C. COMBINING ENSEMBLE MEMBERS

Our choice of ensemble members in Section III-B led us to employee local approach [12] to join the ensemble members. We employed Mixture of Experts (MoE) [13] and Dynamic Selection of Expert (DSE) [14] as local methods to join the ensemble members. The main motivation behind using these methods is our observation that the selected ensemble members are strong learners [12] and their performance varies for different input data. The details of MoE and DSE are given in the following sections.

1) MIXTURE OF EXPERTS

Mixture of Experts is a supervised learning algorithm that harnesses the power of multiple strong learners (that perform the role of experts) weighted through a gating network [13] to generate an overall model with improved accuracy than any of the individual experts. It is a popular method due to its capability of balancing the weakness of some experts with the strengths of others by combining the best approximation parts of the models generated by individual experts to generate the final output.

Using MoE method for execution time predictions of workflows, we aimed at combining execution time predictions of multiple experts to generate the final execution time prediction. Our three tier model of MoE is illustrated in Figure 2. Tier-1 represents input data, which consists of set of workflow attribute values $X = \{x_1, x_2, \ldots, x_n\}$ for which execution time prediction is required. In tier-2, trained experts E_j take input data X and generate their local predictions $p_j = E_j(X, \phi_j)$, where ϕ_j represents set of parameters (learned during training phase) of j^{th} expert E_j.

Tier-3 represents gating network, which decides how the experts will contribute in generating the final prediction. During ensemble training, the gating network was trained using expectation maximization [5], [6] for overall accuracy of the ensemble [14] using training dataset. The derivation of expectation maximization is beyond the scope of this work and for its further details, we refer the reader to [5] and [6]. In summary, it is trained to generate higher weights for the experts with higher accuracy and vice versa. At the time of prediction, the gating network takes X as an input and generates corresponding set of weights $\lambda_j(X, \theta)$ of each $E_j(X, \phi_j)$, where θ represents set of parameters (learned during training phase) of gating network. It is noteworthy that gating network dynamically generates weights for each given input. These dynamically generated weights indicate which expert effectively models the given part of attribute space.

To make the sum of all weights equal to 1, the generated weights are normalized using softmax function [15] as:

$$w_j(X, \theta) = \frac{exp(\lambda_j(X, \theta))}{\sum_{j=1}^{m} exp(\lambda_j(X, \theta))} \tag{1}$$

which brings all weights $w_j(X, \theta)$ in the range of [0, 1].

In tier-4, the weights are combined with experts' local predictions to generate the final execution time prediction $P(X)$ as:

$$P(X) = \sum_{j=1}^{m} w_j(X, \theta) * E_j(X, \phi_j) \tag{2}$$

where m represents total number of experts.

2) DYNAMIC SELECTION OF EXPERT

The dynamic selection of expert (DSE) is similar to MoE that employs multiple experts. However, in place of gating network there is a system that estimates the accuracy of each expert in the vicinity of the given input. It then selects the most accurate expert to generate the final prediction. The algorithm of DSE is shown in Algorithm 1. First, we find Ω as k-nearest neighbors [16] of the given input X (line 1).

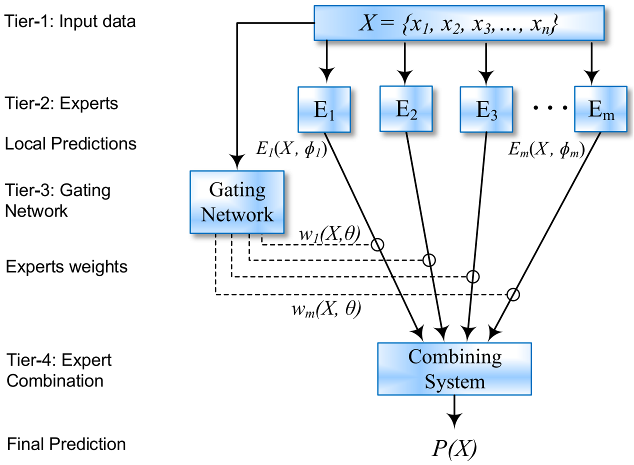


FIGURE 2. Mixture of Experts Model.

The nearest neighbors are found using equation 3, where the distance between two sets of workflow attribute values $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ is calculated as

$$D(X, Y) = \sum_{i=1}^{n} \omega_i \times d(x_i, y_i) \tag{3}$$

Here ω_i represent weight of i^{th} attribute determined by dynamic weighing scheme, as described in [11]. $d(x_i, y_i)$ represents distance between values of i^{th} attribute of X and Y, and

Algorithm 1 Workflow Execution Time Prediction Using Dynamic Selection Of Expert.

```
Require: Set of trained Experts $E = \{E_1, E_2, \ldots, E_n\}$
    $D_t$ : Training dataset
    Set of workflow attribute values $X = \{x_1, x_2, \ldots, x_n\}$
    $k$ : Number of nearest neighbours
Ensure: $P(X)$ : Workflow execution time prediction on $X$
 1: find $\Omega$ as $k$ nearest neighbors of $X$ in $D_t$
 2: for all $E_j \in E$ do
 3:     $A_{sum} = 0$
 4:     for all $X_i \in \Omega$ do
 5:         $p_j = getPredTime(E_j, X_i)$
 6:         $A_{sum} = A_{sum} + calcuPredAccu(X_i, D_t, p_j)$
 7:     end for
 8:     $A_j = A_{sum} / \mid \Omega \mid$
 9: end for
10: $A_{max} = arg\ max_j\{A_j\}$
11: $E_{max} = getCorrespondExpert(A_{max}, A)$
12: $P(X) = E_{max}(X)$
13: return $P(X)$
```

was calculated by adapting the definitions of ''heterogeneous Euclidean-overlap metric'' [7].

In next step, we calculate average accuracy of each expert for Ω (line 2 - line 9). Next, the expert with the maximum accuracy E_{max} is selected (line 11). The final prediction $P(X)$ is given as an output of E_{max} for X, (line 12 - 13).

IV. EXPERIMENTS AND EVALUATION

Our testbed for experiments, the workflows and the results of experiments are described in the following sections.

A. SCIENTIFIC WORKFLOWS

The proposed methods were evaluated for three real world scientific workflows: WaveWatch III [17], Invmod [18], and MeteoAG [19]. The selected workflows vary largely with respect to their execution times, number of parallel and sequential tasks, structures, etc.

WaveWatch III (WW3) workflow produces wave simulations in oceans using discrete spectral action balance equation. National Oceanic and Atmospheric Administration (NOAA) of USA uses WaveWatch III wave model to make operational ocean wave predictions. The WaveWatch III workflow is depicted in Figure 3(a). Invmod workflow aims at analyzing the impact of climate variation on water balance by simulating the water flow and balance model. This workflow also accurately predicts the possibility of extreme floods. Figure 3(b) depicts Invmod workflow. The MeteoAG workflow simulates atmospheric fields of heavy precipitation using regional atmospheric modeling system (RAMS) [20]. These simulations are used to predict thunderstorms and watersheds in an area. Figure 3(c) show MeteoAG workflow and its tasks.

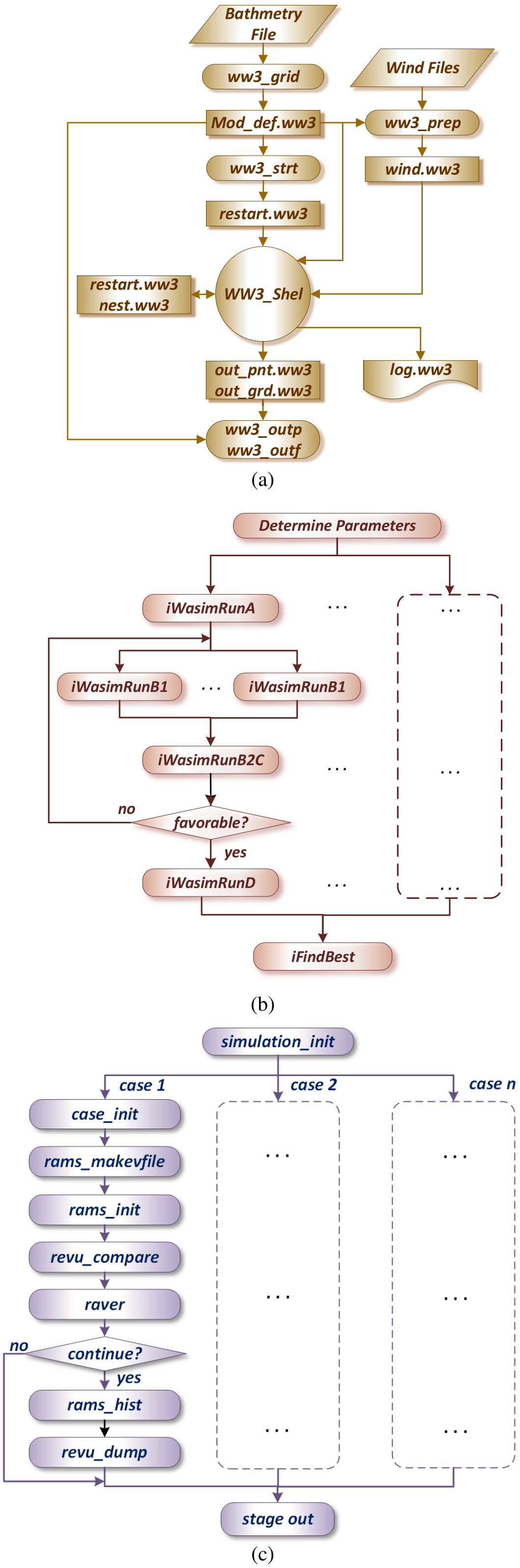
These workflows were run in FCIT Grid environments and Austrian Grid and traces were collected. For the details of the Grid environments, see the following section.

B. TESTBED ENVIRONMENT

We thoroughly evaluated the proposed methods through a series of workflow execution and prediction experiments for two real Grid environments: FCIT Grid and Austrian Grid.

FCIT Grid environment [21] is a dedicated job-based Grid developed for researchers, students and faculty members of King Abdulaziz University to run their scientific experiments. FCIT Grid-sites are custom designed to facilitate needs of local users when they share these Grid-sites. Different resource sharing policies are also implemented. Table 1 briefly describes FCIT Grid-sites and workflow execution experiments on these Grid-sites.

Austrian Grid [22] is Austria's National Grid comprising several heterogeneous Grid-sites located in different cities. These Grid-sites are maintained by different owners who enforce different policies for sharing their resources among Grid users. Some of these Grid-sites are dedicated for Austrian Grid while others are opportunistically available. Table 2 shows a summary of Grid-sites in Austrian Grid and the number of workflow tasks run on these Grid-sites.



(a)

(b)

(c)

FIGURE 3. Scientific workflows considered for experimental evaluation. (a) WaveWatch III Workflow. (b) Invmod workflow. (c) MeteoAG workflow.

The selected workflows (see Section IV-A) were run at different times (days/hours) on different Grid-sites and for various workflow problem-sizes. At the time of experiments,

TABLE 1. Grid-sites from FCIT Grid used in current study and workflow tasks run on these Grid-sites.

Grid-site*	Architecture	Cores	OS	LRM	Workflow Tasks
FCIT-GS1	Intel Xeon E5-2407 (2.20GHz/4c)	40	Linux	SGE	23893
FCIT-GS2	Intel Xeon E5-2440 (2.0GHz/ 6c)	20	Linux	SGE	13325
FCIT-GS3	Intel Xeon E5-2440 (2.0GHz/ 6c)	40	Linux	SGE	32834
FCIT-GS4	Intel Xeon E5-2660v2 (2.20GHz/10c)	30	Linux	Torque	25547
FCIT-GS5	Intel Xeon E5-2660v2 (2.20GHz/10c)	50	Linux	Torque	25946
FCIT-GS6	Intel Xeon E5-2660v2 (2.20GHz/10c)	60	Linux	SGE	15278
FCIT-GS7	Intel Xeon E5-2660v2 (2.20GHz/10c)	100	Linux	Torque/Maui	19448

*We have anonymized the actual names of the Grid-sites.

TABLE 2. Grid-sites from austrian grid used in current study and workflow tasks run on these grid-sites.

Grid-site*	Architecture	CPUs	OS	LRM	Workflow Tasks
AG-GS1	Ethernet Pentium 4	20	Linux	PBS	23963
AG-GS2	SGI Altix 3000 Itanium 2	64	Linux	PBS	21780
AG-GS3	SGI Altix 350 Itanium 2	16	Linux	Torque	24846
AG-GS4	Pentium 4	20	Linux	PBS	14093
AG-GS5	AMD Athelon(tm)	16	Linux	Torque	13434
AG-GS6	AMD Opteron 848	80	Linux	SGE	26115
AG-GS7	AMD Opteron 248	198	Linux	SGE	12188
AG-GS8	Intel Xeon 2.0	12	Linux	Torque	14596

*We have anonymized the actual names of the Grid-sites.

the queues of local resource managers (LRMs) at the Grid-sites were in different states. In a nutshell, the experiments trace dataset comprised many different workflow execution cases.

C. EMPIRICAL EVALUATION

Broadly, our experiments aimed at thorough evaluation of the proposed methods for a variety of real workflows in real Grid environments. Specifically, we designed our experiments to find answers of the following questions.

- How effective is the multi-model approach to predict execution time of workflows as compared with individual models in related work?
- How does dynamic selection of expert perform as compared with mixture of experts for workflow execution time predictions?
- To what extent we can refine accuracy of workflow execution time predictions using multi-model approach?
- How does the proposed approach perform for different scientific workflows in different Grids?

Our findings of the experiments are described in the following sections.

In all our experiments, we divided the trace dataset into distinct training dataset and the test dataset. The execution time of workflow was measured in seconds. The accuracy of proposed methods was evaluated in terms of mean absolute error of m experiments, calculated as:

$$mean\ absolute\ error = \frac{\sum_{j=1}^{m} |t_r^j - t_p^j|}{m} \tag{4}$$

where t_r^j and t_p^j represent real and predicted execution times of j^{th} experiment, respectively. For the sake of comparative evaluations of different sets of experiments mean absolute error was normalized with average real time to calculate mean normalized error (MNE). A second metric of absolute relative error calculated as $|t_r - t_p|/(t_r + t_p)$ was also used to evaluate the proposed methods. It is noteworthy that our evaluation metrics are in accordance with the metrics in [8], [9], and [11].

The proposed methods were evaluated for the selected workflows (see Section IV-A) by varying Grid-sites and workflow problem-sizes in two different Grid environments. These experiments were conducted at different hours of the day and they covered various values of dynamic attributes (it is very hard to choose their values because those are determined by runtime environment). In all our evaluations, we repeated each experiment 10 times (to remove any possible anomalies) and their mean values are shown here.

We found MoE method outperforming DSE method in all experiments (the comparative results are discussed later in this section). Therefore, we are describing here the detailed results of experiments with MoE method only.

Figure 4 shows MNE in predicted execution times through the MoE method for WW3, Invmod, and MeteoAG workflows in FCIT Grid for different numbers of Grid-sites $(1-8)$ and three different problem-sizes. For WW3 workflow, the maximum (minimum) MNE was 11% (5%) with an overall average of 8%. The maximum (minimum) MNE was found when problem-size:1 and Grid-sites $= 1$ (problem-size:2 and Grid-sites $= 5$). For Invmod workflow, the maximum (minimum) MNE was 12% (6%) with an overall average of 9%. The maximum (minimum) MNE was found when problem-size:2 and Grid-sites $= 2$ (problem-size:3 and Grid-sites $= 7$). For MeteoAG workflow, the maximum (minimum) MNE was 10% (4%) with an overall average of 7%. The maximum (minimum) MNE was

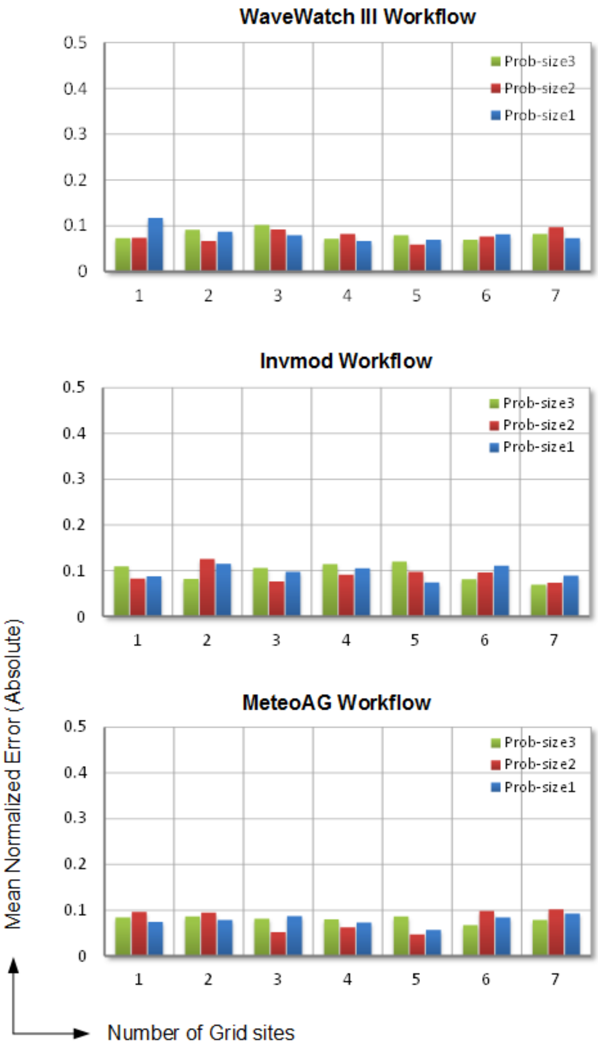


FIGURE 4. Mean normalized error (absolute) for WW3 (top), Invmod (middle), and MeteoAG (bottom) workflows for different number of Grid-sites and problem-sizes in FCIT Grid using MoE method.

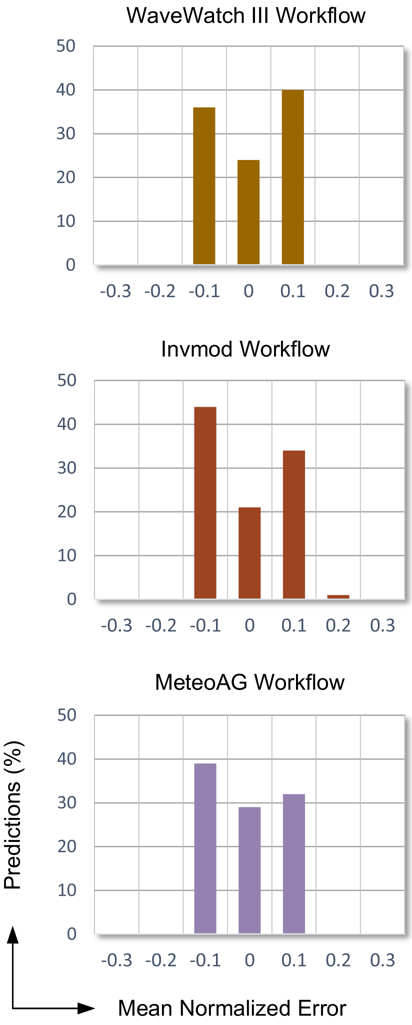


FIGURE 5. Collective distribution of mean normalized error (absolute) for Invmod (top), MeteoAG (middle), and WW3 (bottom) workflows in two Grid environments.

observed when problem-size:2 and Grid-sites = 7 (problem-size:2 and Grid-sites = 5). Figure 5 shows distribution of MNE for the three workflows in both Grids. We did not observe any specific pattern in MNE as we varied problem-size and number of Grid-sites.

Figure 6 shows MNE in predictions through the MoE method for Invmod and MeteoAG workflows in Austrian Grid for three different problem-sizes and different numbers of Grid-sites (1−8). For Invmod workflow, the maximum (minimum) MNE was 10% (5%) with an overall average of 7%. The maximum (minimum) MNE was found when problem-size:3 and Grid-sites = 1 (problem-size:2 and Grid-sites = 7). For MeteoAG workflow, the maximum (minimum) MNE was 11% (4%) with an overall average of 8%. The maximum (minimum) MNE was observed when problem-size:3 and Grid-sites = 6(problem-size:2 and Grid-sites = 5). Once again, no patterns were observed in MNE as we varied number of Grid-sites and problem-size.

The results using absolute relative error were found to be very similar to mean absolute error. Therefore, for the sake of simplicity, we have excluded those results.

We also compared the proposed methods (mixture of experts and dynamic selection of expert) with three methods from related work that use radial basis function neural network (RBF-NN) [9], local learning framework (LLF) [11], and performance templates generated through evolutionary programming (PT-EP) [8] to predict workflow execution time. The motivation behind choosing these methods for our comparison is to select only those methods that support all our selected workflow performance attributes.

Figure 7 depicts a comparison of five prediction methods for WW3, Invmod, and MeteoAG workflows in FCIT Grid. We found MoE outperforming other methods – showing minimum MNE – for all three workflows. For WW3 workflow, MNE was reduced by DSE by 21%, 37%, and 50% as compared with RBF-NN, LLF, and PT-EP, respectively. MoE reduced MNE for WW3 by 33%, 48%, 58%, and 66%,

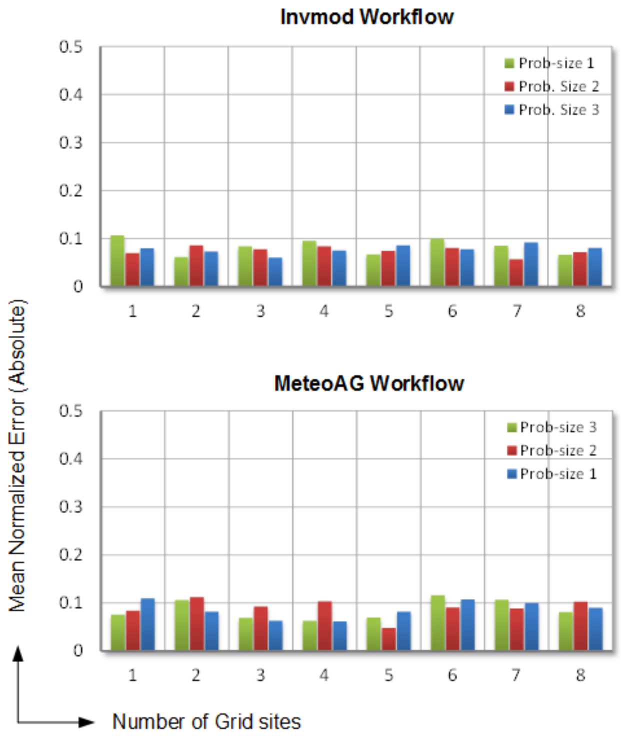


FIGURE 6. Mean normalized error for Invmod (top), and MeteoAG (bottom) workflows for different number of Grid-sites and problem-sizes in Austrian Grid using MoE method.

as compared with that of DSE, RBF-NN, LLF, and PT-EP, respectively. For Invmod workflow, DSE reduced MNE by 32%, 49%, and 57% as compared with RBF-NN, LLF, and PT-EP, respectively. MoE reduced MNE for Invmod by 18%, 44%, 58%, and 65%, as compared with that of DSE, RBF-NN, LLF, and PT-EP, respectively. For MeteoAG workflow, DSE reduced MNE by 25%, 33%, and 47% as compared with RBF-NN, LLF, and PT-EP, respectively. MoE reduced MNE for MeteoAG by 32%, 49%, 54%, and 64%, as compared with that of DSE, RBF-NN, LLF, and PT-EP, respectively.

Figure 8 shows comparison of five prediction methods for Invmod and MeteoAG workflows in Austrian Grid. Again, we found mixture of experts as the best method (with the minimum MNE) for both workflows. For Invmod workflow, the MNE using DSE was reduced by 33%, 51%, and 58% as compared with that of RBF-NN, LLF, and PT-EP, respectively. The MNE using MoE for Invmod was reduced by 29%, 53%, 65%, and 70%, as compared with that of DSE, RBF-NN, LLF, and PT-EP, respectively. For MeteoAG workflow, the MNE using DSE was reduced by 23%, 30%, and 37% as compared with that of RBF-NN, LLF, and PT-EP, respectively. The MNE using MoE for MeteoAG was reduced by 39%, 53%, 57%, and 62%, as compared with that of DSE, RBF-NN, LLF, and PT-EP, respectively.

We also evaluated the proposed methods by varying size of datasets and found that accuracy improved with more data.

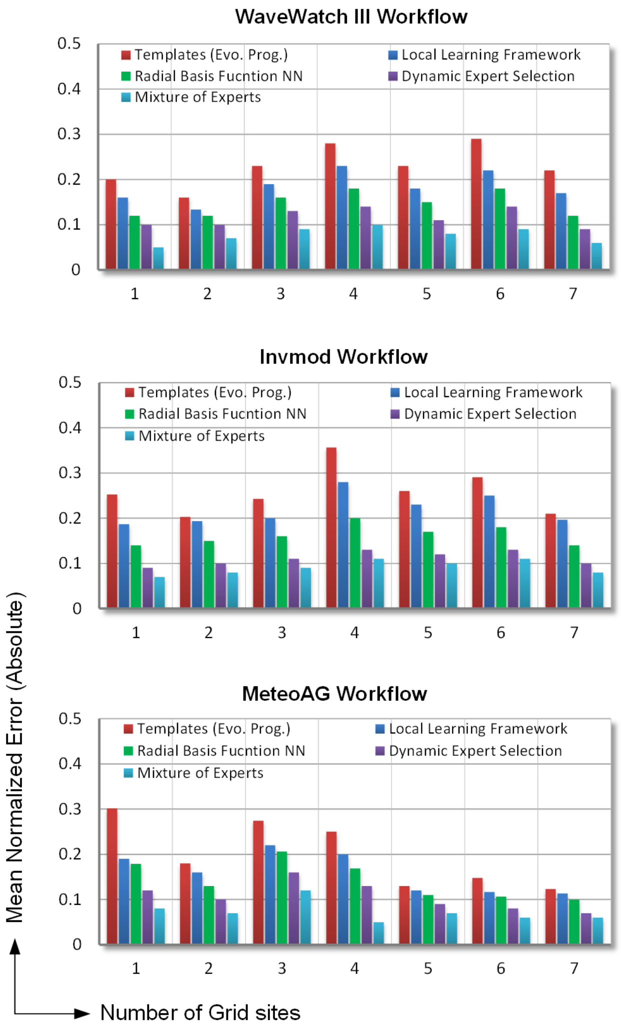


FIGURE 7. A comparison of mean normalized error (absolute) of the two proposed methods with three other methods from related work for Invmod (top), MeteoAG (middle) and WaveWatch III (bottom) workflows for different Grid-sites in FCIT Grid.

We conducted our experiments using Weka toolkit with additional programming for some features. On MacBook Pro 2.3 GHz, quad core, Intel core i7 with 16 GB memory, it took less than 40 seconds to build MoE for each workflow and a few milli seconds (19 ms on average) for each prediction. For DSE it took about 30 seconds in learning phase and a few milli seconds (17 ms on average) for each prediction, which is reasonable and affordable for major clients of prediction service, e.g. workflow scheduling services [23], [24]. It is noteworthy that the training of ensemble is done offline and required only once at the start of the prediction service.

V. RELATED WORK

Execution time of applications comprising single tasks has been predicted using different types of approaches, e.g. based on historical data [25]–[28], statistical modeling [29]–[31], analytical modeling [32]–[34], hybrid modeling using analytical and statistical models together [35], application benchmarking [36]–[38], data mining [27], [39]–[41], time series

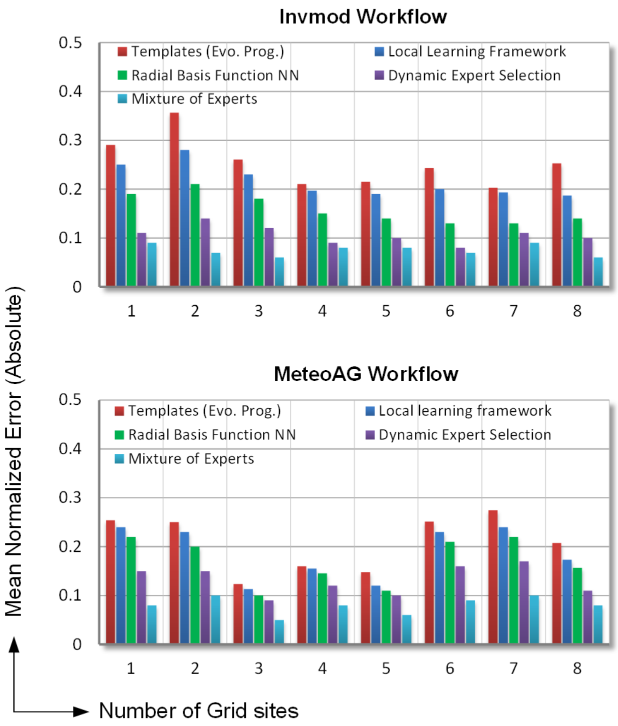


FIGURE 8. A comparison of mean normalized error (absolute) of the two proposed methods with three other methods from related work for Invmod (top), MeteoAG (bottom) workflows for different Grid-sites in Austrian Grid.

analysis [42], [43], partial execution [44], simulation [45], coupling machine profiles with application information [29], [46]–[50], etc. However, these approaches can predict execution time of applications consisting of single task on single Grid-site and cannot model variations in application problem-size. In contrast to these studies, our focus is to predict execution time of e-science workflows considering variations in Grid-sites and problem-sizes.

There have been some studies to forecast execution time of individual tasks in workflows. Miu and Missier [51] predicted execution time of individual tasks in workflows based on historical data and task input features. Liu *et al.* [52] exploited statistical time-series patterns to predict bounds of run time of workflow tasks. da Silva *et al.* [53] employed workflow profiling to model resource consumption and execution time of individual tasks in workflows. Pham *et al.* [54] forecasted run time of workflow tasks in the Cloud using machine learning approach. Their approach is based on parameters defining runtime environment. An incremental learning approach has been adopted by Hilman *et al.* [55] to forecast execution time of individual tasks in e-science workflow applications.

Random variable models were also employed for workflow execution time modeling and prediction. Mussi and Nain [56] and Gelenbe [57] targeted at modeling execution time of task graphs using random variable models. The authors also modeled the distribution of execution time based on graph parameters. Chirken *et al.* [58] used workflow structural information and execution time of individual tasks as parameters to model execution time of the whole workflow using random variable modeling. The authors also developed distribution function of workflow execution time using the model parameters. Glatard *et al.* [59] exploited probabilistic modeling to investigate run time of complete workflows in the Grid. Their approach takes task run times, and data transfer times as input and model the rest part of the entire execution time as a random variable. These methods do not handle variations in execution time due to input problem-size. In addition, these methods consider simple workflow structures and cannot handle complex task dependencies. These methods assume a pre-defined mapping and also assume that the task execution times on the selected Grid-sites are available. Contrary to these methods, our proposed approach does not require execution times of individual workflow tasks and can be affectively used for complex workflow structures as well as variations in input problem-size. Our methods model workflow execution time as a whole considering dynamic assignment of workflow tasks on the Grid-sites. Our predictions are definite quantities (time in seconds), contrary to probabilistic models.

Singh *et al.* [60], [61] used machine learning agents to predict workflow execution time. The agents captured pre-defined performance metrics during execution of a workflow task on a Grid-site. The performance metrics of agents are later combined to develop overall workflow performance metrics.

So far, only Nadeem *et al.* have predicted complete workflow execution time considering all our supported attributes using RBF neural network [9], local learning framework [11], and similarity templates [8], [62]. The authors considered detailed attributes defining workflow execution time at different levels including its structure [63] and execution environment. However, present study proposes two multi-model ensemble systems that combined three models by Nadeem *et al.* to overcome the deficiencies of each individual model and thus improved the overall prediction accuracy. Our proposed methods reduced the prediction error by 50% (on average) as compared with individual models by Nadeem *et al.* Multi-model ensemble systems have been successfully used in several other fields, e.g. numerical weather prediction [64], decadal predictions [65], flood forecasting [66], image classification [67], industrial process monitoring [68] etc.

Some other studies have also addressed workflow execution time prediction in the Clouds. Pietri *et al.* [69] employed number of Cloud resources and workflow structure information to model workflow execution time. Hiden *et al.* [70] used performance models developed from partial execution of the workflow to forecast execution time of the entire workflows in the Cloud. The accuracy of their models is improved as more performance data of the application is available. In another effort, Hiden *et al.* [71] used provenance data to generate multiple models to forecast run times of medical data processing workflows in the Cloud. Pham *et al.* [54] used a two-phase machine learning method to predict run time of workflows in the Cloud.

VI. CONCLUSION AND FUTURE WORK

Workflow execution time predictions are crucial for workflow user, workflow management middleware, as well as the resource owners to take different decisions about workflow execution and optimization. Predicting execution time of complete workflows in the Grid is a hard task due to various static and dynamic factors affecting workflow performance during its life cycle. In this paper, we proposed two novel multi-model machine learning ensemble systems (our first major contribution) mixture of experts and dynamic selection of expert to predict execution time of e-science workflows in distributed environments like Grids. Our employed experts are strong learners by themselves. The mixture of experts method combines local predictions of the experts weighted dynamically through a gating network to generate final execution time prediction. The gating network is trained using expectation maximization for accuracy of the final prediction. The dynamic selection of expert method selects the most accurate expert in the vicinity of the given input to generate the final workflow execution time prediction. The vicinity of the given input is determined in terms of k-nearest neighbors found using heterogeneous Euclidean-overlap metric. We evaluated the proposed methods through a series of experiments for three real scientific workflows in two different Grid environments. The experiments were made for different setups of Grid environments (our second contribution), number of Grid-sites and workflow problem-sizes. The evaluation results were compared with three approaches from the related work. We observed that the proposed approach significantly (50%, on average) reduces the prediction errors when compared with the previous approaches (our third contribution). The mixture of experts was also observed to be more accurate than dynamic selection of expert in our experiments. The proposed approach can be also be effectively used for other heterogeneous distributed environments like the Cloud, if the trace data for the parameters defining workflow performance in the target environment is available.

In future, we plan to evaluate the proposed approach for other scientific workflows in the Cloud. Finding effects of variations in different workflow attributes' values on prediction error is also part of our future plans. We also plan to model workflow performance through probabilistic variable.

ACKNOWLEDGEMENT

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH) - King Abdulaziz City for Science and Technology - the Kingdom of Saudi Arabia - award number (12-INF2716-03). The authors also, acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support.

REFERENCES

[1] E. Deelman *et al.*, "Pegasus, a workflow management system for science automation," *Future Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

[2] T. Fahringer *et al.*, "ASKALON: A development and Grid computing environment for scientific workflows," in *Workflows for E-Science*. London, U.K.: Springer, 2007, pp. 450–471.

[3] N. Kumar and D. P. Vidyarthi, "A hybrid heuristic for load-balanced scheduling of heterogeneous workload on heterogeneous systems," *Comput. J.*, vol. 6, no. 2, pp. 276–291, 2018.

[4] R. Entezari-Maleki, K. S. Trivedi, L. Sousa, and A. Movaghar, "Performability-based workflow scheduling in grids," *Comput. J.*, vol. 61, no. 10, pp. 1479–1495, 2018.

[5] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.

[6] M. I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures," *Neural Netw.*, vol. 8, no. 9, pp. 1409–1431, 1995.

[7] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *J. Artif. Intell. Res.*, vol. 6, pp. 1–34, Jan. 1997.

[8] F. Nadeem and T. Fahringer, "Optimizing execution time predictions of scientific workflow applications in the grid through evolutionary programming," *Future Gener. Comput. Syst.*, vol. 29, no. 4, pp. 926–935, Jun. 2013.

[9] F. Nadeem, D. Alghazzawi, A. Mashat, K. Fakeeh, A. Almalaise, and H. Hagras, "Modeling and predicting execution time of scientific workflows in the grid using radial basis function neural network," *Cluster Comput.*, vol. 20, no. 3, pp. 2805–2819, Sep. 2017.

[10] A. Kaushik and D. P. Vidyarthi, "A green energy model for resource allocation in computational grid," *Comput. J.*, vol. 58, no. 7, pp. 1530–1547, 2015.

[11] F. Nadeem and T. Fahringer, "Predicting the execution time of grid workflow applications through local learning," in *Proc. Conf. High Perform. Comput. Netw., Storage Anal. (SC)*, Nov. 2009, pp. 33:1–33:12.

[12] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2014.

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[14] A. S. Britto, Jr., R. Sabourin, and L. E. S. Oliveira, "Dynamic selection of classifiers—A comprehensive review," *Pattern Recognit.*, vol. 47, no. 11, pp. 3665–3680, Nov. 2014.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Secaucus, NJ, USA: Springer-Verlag, 2006.

[16] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, Apr. 1997.

[17] J. L. Hanson, B. A. Tracy, H. L. Tolman, and R. D. Scott, "Pacific hindcast performance of three numerical wave models," *J. Atmos. Ocean. Technol.*, vol. 26, no. 8, pp. 1614–1633, 2009.

[18] D. Theiner, "Reduction of calibration time of distributed hydrological models," in *Proc. Int. Conf. Hydroinformatics*, Nice, France, Sep. 2006, pp. 4–8.

[19] F. Schüller, J. Qin, F. Nadeem, R. Prodan, T. Fahringer, and G. Mayr, "Performance, scalability and quality of the meteorological grid workflow meteoag," in *Proc. 2nd Austrian Grid Symp.*, Innsbruck, Austria, Sep. 2006, pp. 21–23.

[20] S. M. Saleeby and W. R. Cotton, "A large-droplet mode and prognostic number concentration of cloud droplets in the colorado state University regional atmospheric modeling system (RAMS). Part I: Module descriptions and supercell test simulations," *J. Appl. Meteorol.*, vol. 43, no. 1, pp. 182–195, 2004.

[21] *FCIT Grid at King Abdulaziz University W.R.T to Workflow Performance Predictin Project*. Accessed: Dec. 31, 2018. [Online]. Available: https://fabdullatif.kau.edu.sa/Pages-prj-perMod.aspx

[22] *The Austrian Grid Consortium*. Accessed: Dec. 30, 2012. [Online]. Available: www.austriangrid.at

[23] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, to be published.

[24] H. Chen, J. Wen, W. Pedrycz, and G. Wu, "Big data processing workflows oriented real-time scheduling algorithm using task-duplication in geo-distributed clouds," *IEEE Trans. Big Data*, to be published.

[25] R. Gibbons, "A historical application profiler for use by parallel schedulers," in *Job Scheduling Strategies for Parallel Processing*. New York, NY, USA: Springer-Verlag, 1997.

[26] N. H. Kapadia, J. A. B. Fortes, and C. E. Brodley, "Predictive application-performance modeling in a computational grid environment," in *Proc. 8th Int. Symp. High Perform. Distrib. Comput.*, Aug. 1999, pp. 47–54.

[27] H. Li, D. Groep, J. Templon, and L. Wolters, "Predicting job start times on clusters," in *Proc. CCGRID IEEE Int. Symp. Cluster Comput. Grid*, Apr. 2004, pp. 301–308.

[28] B. Mohr and F. Wolf, "KOJAK—A tool set for automatic performance analysis of parallel programs," in *Proc. Eur. Conf. Parallel Process.* Berlin, Germany: Springer, 2003, pp. 1301–1304.

[29] Q. Wu and V. V. Datla, "On performance modeling and prediction in support of scientific workflow optimization," in *Proc. IEEE World Congr. Services*, Jul. 2011, pp. 161–168.

[30] B. J. Barnes, B. Rountree, D. K. Lowenthal, J. Reeves, B. de Supinski, and M. Schulz, "A regression-based approach to scalability prediction," in *Proc. 22nd Annu. Int. Conf. Supercomput. (ICS)*, 2008, pp. 368–377.

[31] X. Wu, V. Taylor, and J. Paris, "A Web-based prophesy automated performance modeling system," in in *Proc. Int. Conf. Web Technol., Appl. Services (WTAS)*, 2006, pp. 17–19.

[32] D. A. Bacigalupo, S. A. Jarvis, L. He, D. P. Spooner, D. N. Dillenberger, and G. R. Nudd, "An investigation into the application of different performance prediction methods to distributed enterprise applications," *J. Supercomput.*, vol. 34, no. 2, pp. 93–111, 2005.

[33] J. Brehm and P. H. Worley, "Performance prediction for complex parallel applications," in *Proc. 11th Int. Parallel Process. Symp.*, Apr. 1997, pp. 187–191.

[34] E. J. H. Yero and M. A. A. Henriques, "Contention-sensitive static performance prediction for parallel distributed applications," *Perform. Eval.*, vol. 63, nos. 4–5, pp. 265–277, May 2006.

[35] L. Carrington, A. Snavely, and N. Wolter, "A performance prediction framework for scientific applications," *Future Gener. Comput. Syst.*, vol. 22, no. 3, pp. 336–346, Feb. 2006.

[36] A. Tirado-Ramos, G. Tsouloupas, M. Dikaiakos, and P. Sloot, "Grid resource selection by application benchmarking for computational haemodynamics applications," in *Computational Science*, vol. 3514. Berlin, Germany: Springer, 2005, pp. 534–543.

[37] F. Nadeem *et al.*, "Soft benchmarks-based application performance prediction using a minimum training set," in *Proc. IEEE Int. Conf. e-Sci. Grid Comput.*, Dec. 2006, p. 71.

[38] R. Prodan, F. Nadeeem, and T. Fahringer, "Benchmarking grid applications for performance and scalability predictions," in *Handbook of Research on Scalable Computing Technologies*, K.-C. Li, C.-H. Hsu, L. T. Yang, J. Dongarra, and H. Zima, Eds. Harrisburg, PA, USA: IGI Global, 2010.

[39] W. Smith, I. Foster, and V. Taylor, "Predicting application run times with historical information," *J. Parallel Distrib. Comput.*, vol. 64, no. 9, pp. 1007–1016, Sep. 2004.

[40] E. Ipek, B. R. de Supinski, M. Schulz, and S. A. McKee, "An approach to performance prediction for parallel applications," in *Proc. 11th Int. Eur. Conf. Parallel Process.* Berlin, Germany: Springer-Verlag, 2005, pp. 196–205.

[41] B. C. Lee, D. M. Brooks, B. R. de Supinski, M. Schulz, K. Singh, and S. A. McKee, "Methods of inference and learning for performance modeling of parallel applications," in *Proc. 12th ACM SIGPLAN Symp. Princ. Pract. Parallel Program.* 2007, pp. 249–258.

[42] X. Liu, J. Chen, K. Liu, and Y. Yang, "Forecasting duration intervals of scientific workflow activities based on time-series patterns," in *Proc. IEEE 4th Int. Conf. eScience*, Dec. 2008, pp. 23–30.

[43] M. A. Iverson, F. Özguner, and L. Potter, "Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment," *IEEE Trans. Comput.*, vol. 48, no. 12, pp. 1374–1379, Dec. 1999.

[44] L. T. Yang, X. Ma, and F. Mueller, "Cross-platform performance prediction of parallel applications using partial execution," in *Proc. ACM/IEEE Conf. Supercomput.*, Nov. 2005, p. 40.

[45] R. Susukita *et al.*, "Performance prediction of large-scale parallell system and application using macro-level simulation," in *Proc. ACM/IEEE Conf. Supercomput.*, Nov. 2008, pp. 20:1–20:9.

[46] F. Vraalsen, R. A. Aydt, C. L. Mendes, and D. A. Reed, "Performance contracts: Predicting and monitoring grid application behavior," in *Grid Computing*, vol. 2242. Berlin, Germany: Springer, 2001, pp. 154–165.

[47] A. Snavely, L. Carrington, N. Wolter, J. Labarta, R. Badia, and A. Purkayastha, "A framework for performance modeling and prediction," in *Proc. ACM/IEEE Conf. Supercomput.*, Nov. 2002, p. 21.

[48] V. Taylor, X. Wu, J. Geisler, and R. Stevens, "Using kernel couplings to predict parallel application performance," in *Proc. 11th IEEE Int. Symp. High Perform. Distrib. Comput. (HPDC)*, Jul. 2002, p. 125.

[49] J. Cao, S. A. Jarvis, D. P. Spooner, J. D. Turner, D. J. Kerbyson, and G. R. Nudd, "Performance prediction technology for agent-based resource management in grid environments," in *Proc. 16th Int. Parallel Distrib. Process. Symp. (IPDPS)*, Apr. 2001, p. 14.

[50] S. Pllana and T. Fahringer, "Performance prophet: A performance modeling and prediction tool for parallel and distributed programs," in *Proc. Int. Conf. Parallel Process.*, Jun. 2005, pp. 509–516.

[51] T. Miu and P. Missier, "Predicting the execution time of workflow activities based on their input features," in *Proc. SC Companion, High Perform. Comput., Netw. Storage Anal.* Washington, DC, USA: IEEE Computer Society, Nov. 2012, pp. 64–72.

[52] X. Liu *et al.*, "A novel statistical time-series pattern based interval forecasting strategy for activity durations in workflow systems," *J. Syst. Softw.*, vol. 84, no. 3, pp. 354–376, Mar. 2011.

[53] R. F. da Silva, G. Juve, M. Rynge, E. Deelman, and M. Livny, "Online task resource consumption prediction for scientific workflows," *Parallel Process. Lett.*, vol. 25, no. 3, 2015, Art. no. 1541003.

[54] T. P. Pham, J. J. Durillo, and T. Fahringer, "Predicting workflow task execution time in the cloud using a two-stage machine learning approach," *IEEE Trans. Cloud Comput.*, to be published.

[55] M. H. Hilman, M. A. Rodriguez, and R. Buyya. (2018). "Task runtime prediction in scientific workflows using an online incremental learning approach." [Online]. Available: https://arxiv.org/abs/1810.04329

[56] P. Mussi and P. Nain, "Evaluation of parallel execution of program tree structures," *SIGMETRICS Perform. Eval. Rev.*, vol. 12, no. 3, pp. 78–87, 1984.

[57] E. Gelenbe, E. Montagne, R. Suros, and C. M. Woodside, "A performance model of block structured parallel programs," in *Proc. Int. Workshop Parallel Algorithms Archit.* Luminy, France: North-Holland, 1986, pp. 127–138.

[58] A. M. Chirkin, A. S. Z. Belloum, S. V. Kovalchuk, and M. X. Makkes, "Execution time estimation for workflow scheduling," in *Proc. 9th Workshop Workflows Support Large-Scale Sci.*, Nov. 2014, pp. 1–10.

[59] T. Glatard, J. Montagnat, and X. Pennec, "A probabilistic model to analyse workflow performance on production grids," in *Proc. 8th IEEE Int. Symp. Cluster Comput. Grid*, Lyon, France, May 2008, pp. 510–517.

[60] A. Singh, M. Nguyen, S. Purawat, D. Crawl, and I. Altintas, "Modular resource centric learning for workflow performance prediction," in *Proc. 6th Workshop Big Data Anal., Challenges, Opportunities (BDAC) and 27th IEEE/ACM Int. Conf. High Perform. Comput., Netw., Storage, Anal. (SC)*, Nov. 2015.

[61] A. Singh, A. Rao, S. Purawat, and I. Altintas, "A machine learning approach for modular workflow performance prediction," in *Proc. 12th Workshop Workflows Support Large-Scale Sci.* New York, NY, USA, 2017, pp. 7:1–7:11. [Online]. Available: http://doi.acm.org/10.1145/3150994.3150998

[62] F. Nadeem and F. Thomas, "Using templates to predict execution time of scientific workflow applications in the grid," in *Proc. 9th IEEE/ACM Int. Symp. Cluster Comput. Grid*. Shanghai, China: IEEE Computer Society, May 2009, pp. 316–323.

[63] A. Wombacher and M. Rozie, "Piloting an empirical study on measures forworkflow similarity," in *Proc. IEEE Int. Conf. Services Comput.*, Sep. 2006, pp. 94–102.

[64] J. Liu, L. Chuanzhe, T. Jiyang, Y. Fuliang, and Y. Wang, "Self-correcting multi-model numerical rainfall ensemble forecasting method," U.S. Patent 20 170 261 646 A1

[65] D. Choudhury, A. Sen Gupta, A. Sharma, R. Mehrotra, and B. Sivakumar, "An assessment of drift correction alternatives for CMIP5 decadal predictions," *J. Geophys. Res., Atmos.*, vol. 122, no. 19, pp. 10,282–10,296, 2017.

[66] L. Liu and Y.-P. Xu, "Impacts of calibration strategies and ensemble methods on ensemble flood forecasting over Lanjiang basin, Southeast China," in *Proc. 19th EGU Gen. Assem. Conf.*, 2017, p. 5840.

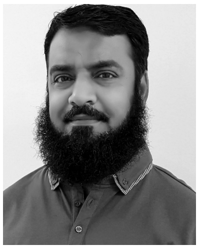
[67] S. Corchs, E. Fersini, and F. Gasparini, "Ensemble learning on visual and textual data for social image emotion classification," *Int. J. Mach. Learn.*, vol. 8, no. 5, pp. 1–14, Oct. 2017.

[68] D. Zurita, M. Delgado, J. A. Carino, and J. A. Ortega, "Multimodal forecasting methodology applied to industrial process monitoring," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 494–503, Feb. 2018.

[69] I. Pietri, G. Juve, E. Deelman, and R. Sakellariou, "A performance model to estimate execution time of scientific workflows on the cloud," in *Proc. 9th Workshop Workflows Support Large-Scale Sci.*, Nov. 2014, pp. 11–19.

[70] H. Hiden, S. Woodman, and P. Watson, "A framework for dynamically generating predictive models of workflow execution," in *Proc. 8th Workshop Workflows Support Large-Scale Sci.*, 2013, pp. 77–87.

[71] H. Hidan, S. Woodman, and P. Watson, "Prediction of workflow execution time using provenance traces: Practical applications in medical data processing," in *Proc. IEEE 12th Int. Conf. e-Sci. (e-Sci.)*, Oct. 2016, pp. 21–30.



FARRUKH NADEEM received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the University of Punjab, Pakistan, the M.B.A. degree from Islamia University, Pakistan, and the Ph.D. degree (Hons.) in computer science from the University of Innsbruck, Austria, in 2009. He has set up the Grid Computing Infrastructure at the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, where he is currently an Associate Professor with the Faculty of Computing and Information Technology. He has been involved in several Austrian research projects and is working on a couple of Saudi research and development projects. He has gained professional trainings on cloud computing and high-performance computing. He has authored more than 29 conference and journal research papers, including four book chapters. His main research interests include performance modeling and prediction, distributed systems, particularly the grid and the cloud, and decision support systems. He is a Member of the program committees of several conferences and an Editorial Board Member of the *Journal of Modern Education and Computer Science*. He holds several distinctions and awards during his educational career. He received the President (King Abdulaziz University) Certificate of Appreciation and a cash award for one of his journal publications.

DANIYAL ALGHAZZAWI received the bachelor's degree (Hons.) in computer science from King Abdulaziz University, in 1999, the master's degree in teaching and leadership from the University of Kansas, in 2004, which helped him to develop his teaching and leadership skills, and the master's and Ph.D. degrees in computer science from The University of Kansas, USA, in 2007. He also received the Certificate of Management International Leadership. He has been the Head of the Information Systems Department, Faculty of Computing and Information Technology, for over five years, during which he organized many workshops and international and domestic conferences. He is currently the Vice dean of Development of the Deanship of IT. He is currently the Head of the Information Security Research Group, King Abdulaziz University. In 2010, he joined University of Essex, and since then, he has published with them several researches in the field of smart environment. In 2017, he got promoted as a consideration for his output in research field which exceeded 90 researches and books in the field of intelligent systems and information security. He has three pending patents. His research interests include smart e-learning, information security, and computational intelligence. He has served as a Reviewer and an Editor for international journals, workshops, and contests.

ABDULFATTAH MASHAT received the B.Sc. degree from King Abdulaziz University (KAU), Jeddah, Saudi Arabia, in 1989, and the M.S. and Ph.D. degrees in computer science from the University of Leeds, U.K., in 1999. He came to his current position after spending three years as the Vice-President for Development at KAU, where he has been the Director of the Information Technology Center, for two years. He was also appointed as the Dean of the Faculty of Computer and Information Technology. During his tenure as the College Dean, he led successful effort for ABET accreditation of the Faculty's three programs. Prior to his Deanship position, he spent several years as the Dean of Admission at the University and was instrumental in the development and implementation of a sophisticated online electronic system, called On Demand University Service (ODUS Plus). He is currently the President of the University of Jeddah, Jeddah. He is developing and implementing key performance indicators for multitudes of university activities. He has authored more than 55 conference and journal research articles. His research interests include performance assessment of asynchronous transfer mode networks, algorithms and protocols for multimedia systems, and models for video traffic VBR. He is a member of several professional and scientific organizations, including ACM. He serves on the governing boards of several Saudi universities and on the advisory boards of several national and international ranking systems.

KHALID FAQEEH received the B.S. degree in computer science from Northrop University, in 1985, and the M.S. and Ph.D. degree in computer science from The George Washington University, in 1990 and 1993, respectively. He has also completed several short courses and trainings in several fields, including leadership management, information technology strategy, distance learning, effective personal productivity, leadership, and so on. He is currently a Professor with the Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. He has been the Head of the Department of Information Systems for eight years and the Dean of the Faculty of Computing and Information Technology for five years. He also has been the Adviser to Ministry of Information, Jeddah. He has over 20 years of experience in expert and decision support systems with a special interest in ethics and computing and the socio-psychological aspects of information systems, as well as media perceptions and cultural and organizational aspects of information Systems. He has authored more than 50 conference and journal publications. He is an Editorial Board Member of the *International Arab Journal of Information Technology* and *International Journal of Information Technology and Web Engineering*.

ABDULLAH ALMALAISE received the B.S. degree in computer science from The University of Southern Mississippi, USA, in 1990, the Master of Management Information Systems degree from the University of Illinois at Springfield, IL, USA, in 1992, and the Ph.D. in computer science from The George Washington University, USA, in 2003. He has 20 years of experience in education field, mainly in teaching and curriculum building of several courses. He has worked on many administrative posts in educational institutes. He was the Chairman of the Management Information Systems Department, College of Business Administration, King Abdulaziz University, Jeddah, Saudi Arabia, from 2005 to 2008. From 2010 to 2012, he was the Chairman of the Computing and Information Technology Department, Jeddah Community College. He was the Chairman of the Department of Information Systems, from 2012 to 2014. He was also the Chair of General Studies at the Computer Science Department, Technical College of Jeddah. He is currently the Vice Dean of Graduate Studies and Scientific Research at the Faculty of Computing and Information Technology, King Abdulaziz University. He is also the Head of the Computer Skills Department. He has expertise in training and supervising the team to learn technical, research, and academic activities. He has more than 45 conference and journal publications and two books. Three of his books are under publications. His research interests include collaborative software, distributed systems conflict measurements, e-Business and e-Government, information systems, and artificial intelligence.

• • •